September 1983 14 HI RES COMPUTER 17 ORIC SIG CONDITIONER

AN EMAP PUBLICATION **75p**

21 PT2 FLEX

26
6502
2MHZ
1MHZ

34 PT2
DIGITISER

40 42 COMPUTER KIT 43 A/D CONVERT 46 ZX VIDEO INFO 55 CIRC BOARDS 61 EXT. CONTROLS

·43
SOUND
BOARD

76 MONITOR
SURVEY

78 83 IC DATA

Electronics & Computing

MONTHLY

Germany Dm 5
United States $2.95

Britain's First Electronics & Computer Applications Magazine

ZX VIDEO TECHNIQUES –

High quality from the '81 and Spectrum

MONITORS – Comprehensive buyers' guide to hi-res displays

DUAL BEAM 'SCOPE
An in-depth review

BBC TIMER – Build a versatile darkroom timer

DRAGON SOUND
The mysteries explained

SIGNAL CONDITIONER
Solve Oric loading problems

REVIEWS
The up market
Apple II E Plus
The low cost **Comx**

LASER
The first £70 colour computer

Electronics & Computing Monthly
Scriptor Court,
155 Farringdon Road,
London,
EC1R 3AD

EDITORIAL
01-833-0846
Editor
Gary Evans
Editorial Assistant
Ann Houghton
Administration
Liz Gregory

ADVERTISING
01-833-0531
Manager
Claire Fullerton
Deputy Manager
Alun Evans
Classified
Nik Saha

PRODUCTION
01-833-0846
Design
Pat Haylock
Terry Gibbins
Make-up
Time Graphics

Publisher
Alfred Rolington

Distribution
EMAP
National Publications

Published By
EMAP Business And
Computer Publications

Printed by
EMLP Peterborough

Subscriptions
Electronics &
Computing Monthly,
(Subscription
Department),
Competition House,
Farmdon Road,
Market Harborough,
Leicestershire.

ABC
MEMBER OF THE AUDIT
BUREAU OF CIRCULATION

IN THIS ISSUE

STOP PRESS ... SEE PAGE 10 FOR MICRODRIVE REPORT

The coupon for our Micronet 800 competition is on page 47.

Want to work for E&CM? – see page 54.

Electronics & Computing Monthly is normally published on the 13th day of each month

EDITORIAL

The Spectrum Microdrive was launched at a gathering of the computer and national press in the last few days of June.

Interest in the drive has been kept alive by various sections of the computer press with a series of 'informed' guesses as to the exact format of the drive. Without this continual hype, its possible that the Microdrive would have faded from the computer aware public's mind. This may have no bad thing as it would have removed the Spectrum's undoubted selling advantage that the promise of the £50 drive bestowed upon the machine.

This is not to say that the Spectrum would not have been a bestseller without the prospect of a cheap, fast access, mass storage system – for many people the Spectrum simply cannot be beaten at its £100 price, more than a year after its launch.

The launch of the Microdrive is bound to give a boost to sales of Sinclair's colour computing but, as the Company's advertising, with an unusual amount of candour, suggests it could be some time before the drives are generally available.

First in the queue to buy the drive will be those early purchasers of the Spectrum. Assuming that the offer of a £50 Microdrive will prove irresistible to a large number of Spectrum owners, getting through the backlog will mean that anyone buying a Spectrum today will have to go to the back of a queue that would stretch a good part of the way from London to Birmingham.

Assuming that the Microdrive suffers from none of the initial production problems that bedevilled the Spectrum our guess is that it will not be until after Christmas at least before the drives are generally available.

All this is not meant to detract from the achievements of Sir Clive and his boys but more a wish that he could have attached enough importance to the Microdrive project to ensure that it met its official launch date of the summer of '82.

New Look E&CM

As from next month *E&CM* will look a little different. As part of our policy of improving the look and feel of the magazine we'll be changing the layout of our front cover.

Another, unavoidable change, will be that, as a result of ever increasing production costs, we are having to put our cover price up to 85p.

We hope you'll like the changes and agree that *E&CM* will still be excellent value for money.

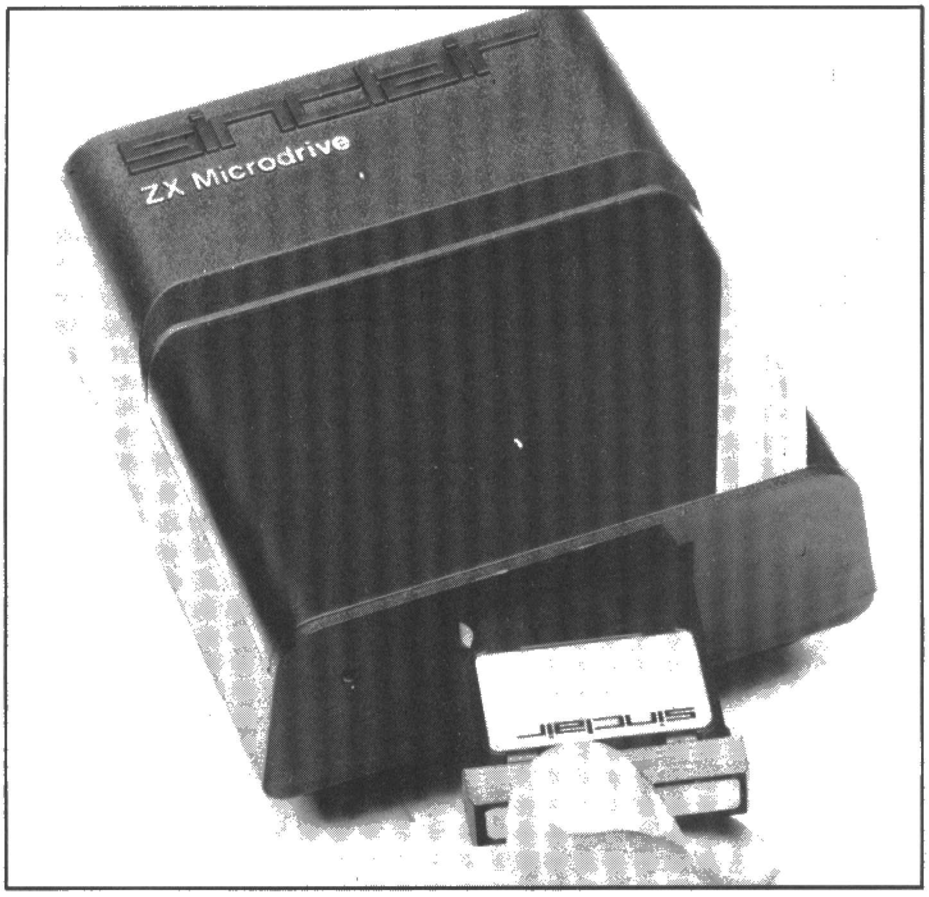
The increase will also mean that those of you with a subscription will reap the benefits of their ways. For those without a subscription, if you're quick enough, you can still get in at the present rate if you can fill in the form now.

Gary Evans

STOP PRESS . . .

The Microdrive Revealed

The wraps have finally been taken off the microdrive. It's been worth waiting for!



The first prototype of the Microdrive was shown over a year ago at the launch of the Spectrum and since then there has been much speculation as to the exact form that the storage media would take. A mini floppy was a favourite at one time, only to give way to the idea that it would be a micro cassette as used in the Hobbit floppy tape system for the BBC micro.

In fact, the Microdrive uses "cartridges" reminiscent of the long dead eight track cartridge system. Sinclair's cartridges are much smaller though, and measure 30mm x 40mm x just 5mm deep. They contain an endless loop of tape that is about 500cm long and 1.9mm wide.

The tape runs at approximately 30 ips when the drive is activated and the recording standard is very similar to that employed in a single density floppy disc system.

The recording head appears to be very similar to that of an audio cassette recorder – something that's not out of the question as the recording density is about 500 bits per inch.

The drive itself is about 8 x 9 x 5cm in size and has a LED indicator to show that it is active.

Interface Details

In order to use a Microdrive with the Spectrum it is necessary to use the ZX Interface 1, a wedge-shaped box that sits under the Spectrum and plugs into the computer's expansion port. The Interface is securely fixed to the Spectrum by two screws – Sinclair have no wish to repeat the problems of wobble that bedevilled the ZX81 RAM Pack.

The Interface contains an 8K ROM in which the software to control the Drives is resident. In addition the Interface features a ULA and some additional circuitry to convert TTL signals to the levels required by the RS232 standard. The Interface also endows the Spectrum with a networking capability.

In Use

Cartridges should not be in place either when power is first applied to the drives or when they are powered down – the manual warns that damage may result if these precautions are not taken.

The first thing to do when using a new cartridge is to issue a FORMAT command. This wipes the cartridge and configures it for use – note that as this command wipes all information from the cartridge it should be used with care. It is possible to write protect a cartridge by removing a small tab which can be covered at a later date to allow writing and reformatting.

The general form of the FORMAT command is

FORMAT "m";d;"name"

where d is the device number – up to 8 Microdrives can be used in tandem – while "name" is a global title that is assigned to a particular cartridge. The FORMAT command is also used to set up the baud rate of the RS232 driver and the first parameter of the command, m in this case, determines for which purpose FORMAT is being used.

The save procedure is quite straightforward and takes the form

SAVE★"m";<drive number>;
"<name>"

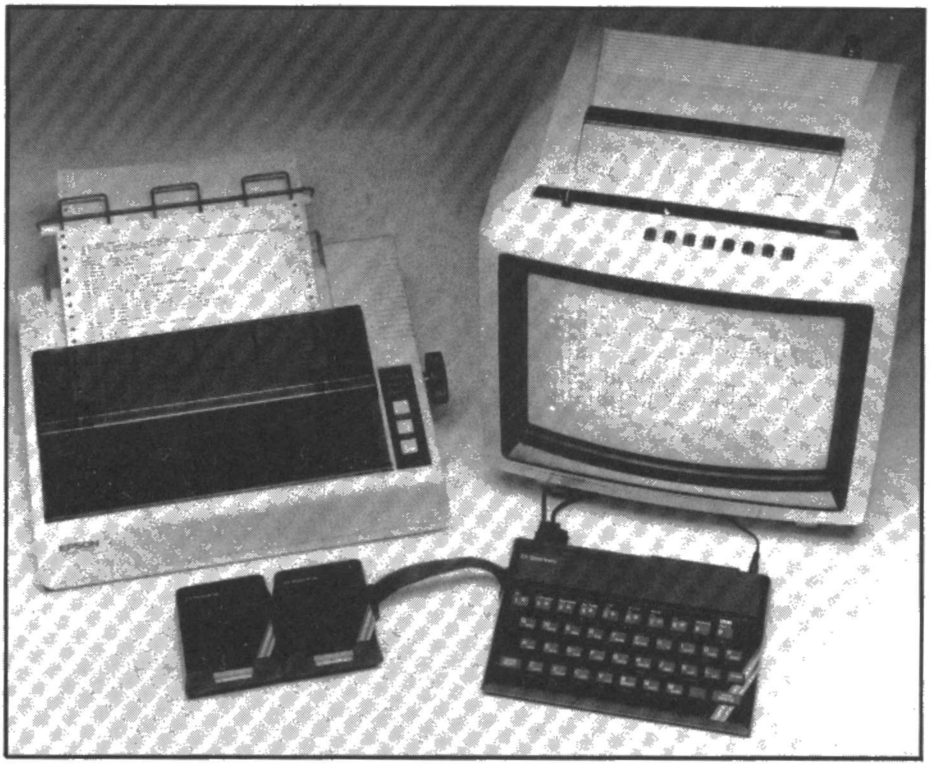
as with cassette storage, programs, screens, arrays and bytes can all be SAVEd and LOADed although, unlike the cassette system, null string files cannot be loaded. Another point to note, and one that will please Software Houses, is that a program SAVEd such that it auto executes cannot be MERGED.

CAT<device number>

will display all the files on a cartridge, although any file with a name beginning with CHR$0 will not appear in this list, another security feature. The CAT command will also show how much of the 100K storage on the cassette is available. In fact the minimum amount of storage guaranteed on any cartridge is 85K but on average 100K is likely to be available.

The Microdrive does not keep a separate file directory on each cartridge, the CAT command merely reads all the file headers on a particular cartridge.

File handling is possible with the Microdrive.



OPEN#<stream number>;"m";
<drive number>;"<name>"

permissible stream numbers range from 0 to 15 and when opened a file can be written to or read from by using one of the following:

PRINT#<stream number>

or

INPUT#<stream number>

for reading variables or

INKEY#<stream number>

to read a stream a byte at a time.

How Fast?

When comparing the Microdrive to a disk system it has a slow access time, average 3.5 seconds, but the data transfer rate of 16K byte per second matches or betters a typical disk system. Against a cassette system there is little comparison.

RS232 Details

The Interface 1 supports two types of RS232 file – a text file ("t") and a binary file ("b").

To open an RS232 file the command form is

OPEN#<stream number>;f$

where f$ is a single character file name. The baud rate is selected by using the FORMAT command as in

FORMAT"t";<baud rate>

The RS232 output is via a 9 pin D-type socket.

Networking

The network capability that the Interface provides allows up to 64 Spectrums to be interconnected via two-core cable terminated in 3.5mm jack plugs. The data transfer rate is about 100K baud but this speed is achieved at the expense of handshaking.

In order to pass a file between two Spectrums, the receiving computer is set up by the command

LOAD★"n";<device number>;
"<file name>"

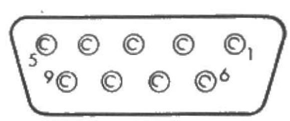
and the Spectrum transmitting the data would use the following

SAVE★"n";<device number>;
"<file name>"

RS232 connections

The RS232 socket is wired as follows:

1. No connection
2. TX data (input)
3. RX data (output)
4. DTR (input) this should be high when ready
5. CTS (output) this should be high when ready
6. n.c.
7. Ground (pull down)
8. n.c.
9. +9v (pull up)



And There's More

This brief look at the Microdrive has not been able to examine the full capabilities of the system. Additional commands such as MOVE (to copy data between two drives via the Spectrum's RAM) and VERIFY are also supported.

For further details of these and more on the technical aspects of the drive read the full review in next month's *E&CM*.

In Closing

There can be little doubt that once again Sir Clive can claim to have entered the market with a product that is far ahead of any competition. At £80 for the drive and interface the Microdrive will bring a flexible mass storage system within reach of a vast number of computer users who just could not afford a £400 disk drive.

It is bound to boost sales of the Spectrum, as if that were needed, and send a legion of peripheral manufacturers to their workbenches in an attempt to adapt the drive for other computers.

The only problem is, that with the target production of only 20,000 Interfaces a month yet to be achieved, and over 500,000 Spectrum owners waiting to get their hands on the drives, it could be some time before they are freely available.

E&CM

Joy at Pifco

Pifco, a familiar name in the British Electroc rather than electronic industry, has recently formed a wholly owned subsidiary to trade under the name Consumer Electronics Ltd.

This new company will in future market a wide range of electronic products but has chosen to launch its marketing endeavours with a comprehensive range of joysticks. Consumer Electronics are not at this stage involved in manufacturing products and the Joysticks are imported Sumcom products.

The range's lowest cost joystick is the Slik Stick, a robust Atari compatible Joystic that retails at the very competitive price of £9.95. Moving up the range, at £13.95 the Starfighter stick provides much the same facilities as the Slik Stick but with a more delux appearance. Both these sticks have been designed for use with Atari computers, the Commodore 64 and Vic 20 machines and, by using an optional adaptor, with the TI 99/4A. In addition, they can be used with an inteference that feature Atari type connectors.

The Starfighter stick is designed for the Apple IIE and, as such, has a suitably up-market price of £39.95. In addition to the basic facilities of the other two controllers, the Starfighter has both left and right fire buttons as well as a third button located on the stick's front edge. It has a centering adjustment and an adjustment to control the throw of the stick. In addition, there is a hi/lo sensitivity switch in the base of the unit.

The TAC 2 is the top of the range as far as conventional Suncom joysticks go and is built to almost arcade standard with, for example, a steel rather than plastic shaft. It is priced at £18.95.

For £29.95, it is possible to break the 'traditional mould of joysticks' with a Joysensor. This provides touch sensitive areas that correspond to the shaft movements of 'normal' sticks. This design may take some getting used to but it is an interesting departure in the art.

The sticks are due for launch in late summer and Consumer Electronics have their eyes firmly set on the lucrative Christmas market.

Review Revisions

A number of points regarding our Joystick reviews las month have been brought to our attention.

With regard to Cambridge Computing's intelligent joystick, the Company have asked us to point out that the fire buttons on the stick are independent, unlike some other designs and that the prices quoted in the article are exclusive of VAT. They would also like to point out that they market the only software programmable stick which features 1K of on board memory and this, with the software supplied on tape, enables it to work with all software.

Atari UK have also asked us to clear up any confusion that may have resulted in a reference to an 'Atari' stick under the review of Voltmace's products.

The reference was to an Atari format stick and not to any product manufactured by Atari themselves.

Finally our apologies to Midwich who were omitted from the review. The Company, together with Flight Link, jointly own the mould from which many of the casings used on British produced analogue type joysticks are produced. The Company's BBC compatible joystick retails at £11.30 + VAT, making it one of the cheapest available.

Games Champ

The 1983 Computer and Video Games Championships reached its final stages at London's Zenon Club last month. Through its columns, *Computer & Video Games* magazine had organised a competition that selected the top 18 Arcade Game players in this country from an initial field of over 500.

The final saw each of the contestants battling it out on six arcade machines before the three highest scorers went on to play a new machine, Gyruss, that was unvieled at the contest.

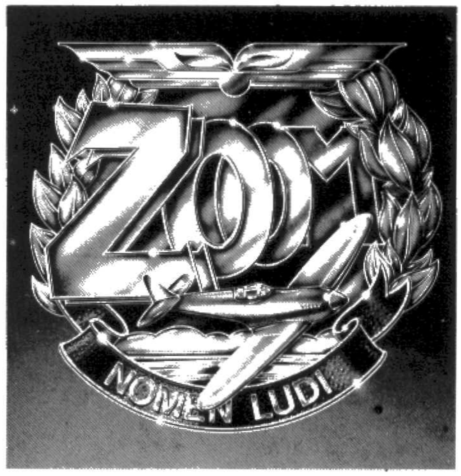
After a few minutes of familiarisation with Gyruss, each player had ten minutes to battle it out on the impressive machine centre stage.

The winner was Julian Rignall, an 18 year old, who, lucky man, keeps the £5000 arcade machine as his prize.

Graphic Update

The design for the *E&CM* Hi-Res computer's graphics board has undergone a number of revisions since it was first published earlier this year. Readers contemplating the construction of this board should send a large SAE to our offices (mark envelope Graphic Board Circuit Diagram) and we will send a copy of the revised diagram.

In future copies of our high-res reprints will include the amended diagram.



Imagine's latest

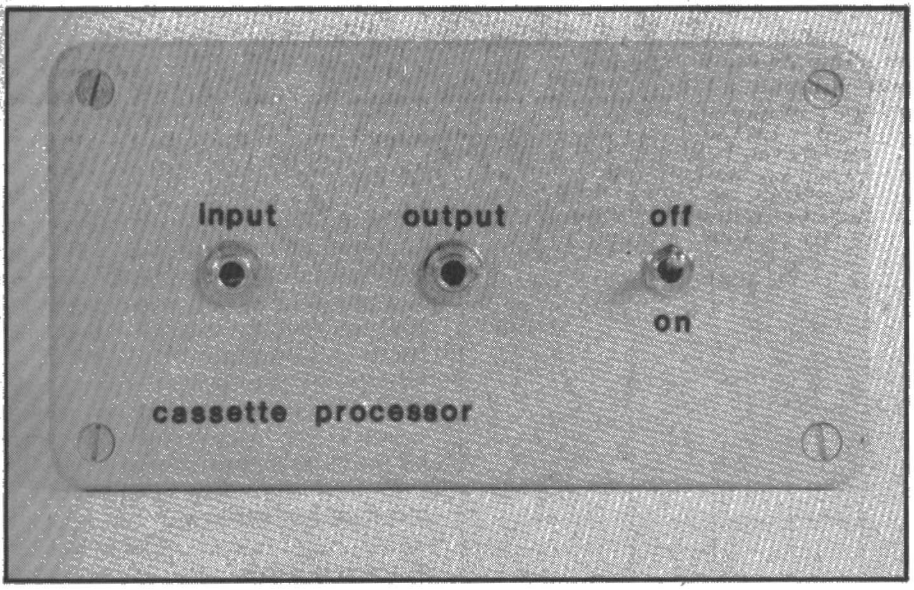
Imagine Software have just released two games for the 48K Spectrum. Both are arcade-action games, with Zip-Zap revolving around a robot sent to prepare an unexplored planet for colonisation. After much laser firing and energy gobbling, the skilful player will achieve the aim of making the outworld fit for human beings.

Zzoom comes complete with a latin motto – nomen ludi, which translates to 'the name of the game'. This is a multi-level space war game wherein the player has to protect refugees from an enemy hell bent on destroying all life.

Both games are up to Imagine's high standards and make full use of the Spectrum's graphics capabilities.

Coinciding with the launch of these new games, Imagine have introduced a new form of packaging for their whole range. The new design can easily be recognised as it features a cardboard box, complete with colourful artwork, in which the game cassette nestles.

Imagine are also committed to maintaining their prices for all games at the modest level of £5.50, at least for the time being.



Cassette Signal Conditioner

Although designed as a result of problems loading programs into the Oric at 2400 baud, Robert Penfold's signal conditioner will be of use with other computers that suffer similar problems.

This processor was designed for use with an Oric 1 computer loading at the fast (2400 baud) rate. Some form of signal processing was considered necessary when attempting fast loads as results with a number of cassette recorders were varied. Some machines were capable of giving adequate results provided the volume control setting was exactly right during replay, while others loaded no more than the first line of a program even with the volume control set at the optimum position. No problems were encountered when loading programs at the slow rate, but at 300 baud long programs take several minutes to load.

After several attempts to improve loading reliability the final, straightforward circuit was evolved. The unit consists of a Schmitt trigger having variable hysteresis, and it is connected between the output of the cassette recorder and the input of the computer. When used with a Philips N2233 cassette recorder (with added earphone socket) saving and loading programs at the fast rate is as straightforward and reliable as when using the slow rate, whereas it had previously been impossible to use the fast rate at all with this recorder.

Although the processor has been specifically designed for use with the Oric 1 computer, it could well be of use with other computers where loading problems are experienced.

On The Slant

Problems with loading programs from cassette can be due to a variety of causes. The use of a good quality cassette tape seems to be well worthwhile, and keeping the tape heads clean will minimise loading troubles. Although most cassette saving systems seem to have some form of tone-leader to set automatic recording levels properly just before saving commences, it still seems to be better to set the recording level manually where this is possible.

It does sometimes happen that a recorder will save and load programs without any difficulty, but will not perform properly if it is used to load a program recorded on another machine. The most likely cause of this is that the azimuth setting of the record/replay head is incorrect. In other words, the recording is being made too high up or too low down on the tape, but this is not likely to matter when the machine is playing back its own recordings since it will scan exactly the same are of tape on which the recording was laid down. The situation is different when a recording made on another machine is played back, with the active part of the head being aligned with only part of the section of tape where the recording is present.

Inaccurate azimuth adjustment results in a loss of high frequency response and virtually guarantees that acceptable results will not be obtained when loading programs you have not recorded yourself. The record/replay head is normally mounted using a simple screw and compression spring arrangement, and one way to correct the screw's setting is to adjust it for maximum treble response when playing back a recording made on a correctly aligned recorder.

Distortion Sources

If the recorder is working as well as could be reasonably expected and reliable loading is nevertheless impossible, then some form of signal processor is probably the only answer. It is unlikely that processing the signal from the computer when saving will be of any use, since the output from the computer can reasonably be assumed to be within specification. One ploy that can sometimes be of benefit is to use a variable (volume control style) attentuator to reduce the output of the computer to a level which is just sufficient to fully drive the recorder. This effectively eliminates the automatic recording level circuit of the recorder and any problems this might be causing.

A lack of high frequency response from the recorder could easily produce loading difficulties, and a high frequency boost circuit could be used in an attempt to improve matters. This, however, could give a high noise level on the output signal and might not be very effective in practice. In any case, provided the recorder is set up correctly and a suitable tape is employed, there should be no serious lack of high frequency performance even if an inexpensive recorder is used.

Phasing distortion may also be a cause of loading problems. This is where some frequencies lag behind others, and is something which is not usually considered to be of paramount importance in audio equipment since no new frequencies are generated by this type of distortion. Despite the fact that this type of distortion may not produce any obvious change in the audio quality of the signal, phase changes which have a non-linear relationship to frequency can produce quite severe waveform distortion and this can be important when a cassette system is used for digital storage and retrieval.

It is possible to use a phase shifter to roughly correct any phase distortion, but reviewing the output from several cassettes on a 'scope while playing back fast loaded Oric programs did not however show any serious irregularities in the waveform. Certainly nothing that would prevent proper ▶

loading, and processing the signal with a variable phase shift circuit did not seem to improve performance discernably.

Final Solution

Using a Schmitt trigger to process the signal is perhaps not an obvious choice since the Oric 1 has a built-in circuit of this type as part of the cassette interface. Adding the Schmitt trigger circuit of **Fig 1**, however, gave an immediate end to loading problems. All a circuit of this type does is to produce a high output state if the input signal is above a certain voltage, or a low output state if it is below a certain voltage. The two threshold voltages are not the same, the first being higher than the second, so that the circuit has a reluctance to change state (an effect known as hysteresis). If a single threshold voltage was used there would be a tendency for noise on the input signal to cause multiple operation of the trigger as the signal passed through the threshold level, and instability in the trigger circuit could also be a problem.

Figure 2 illustrates the way in which the signal is improved by a trigger circuit. Although the input signal is rather irregular, does not have particularly fast rise and fall times, and is quite unlike a logic circuit signal, the output switches "cleanly" between the two logic levels to give a good quality square or pulse output signal. The degree of hysteresis used is unlikely to be critical, but an inadequate amount could produce spurious output pulses due to noise on the input signal, and an excessive amount would make the circuit insensitive to the point where a sufficiently high input level could not be achieved.

The circuit of **Fig 1** is a straightforward

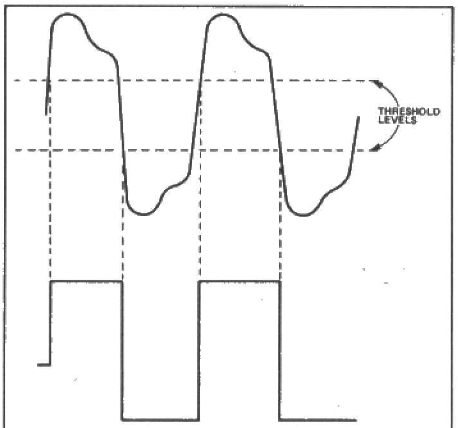


Figure 2. The waveform shown above represents the unprocessed signal from the Oric. The effect of the conditioner is to clean this signal up.

operational amplifier trigger circuit which uses IC1 as a comparator. Its output will be high if the inverting input (pin 2) is at a higher voltage than the non-inverting input (pin 3), or low if the comparative input levels are reversed. R1 and R2 bias the inverting input to half the supply voltage, and the input signal is coupled to this point in the circuit by C1. If we ignore R5 and R6 for the moment, the non-inverting input of IC1 is also biased to half the supply voltage (by R3 and R4). When the input signal is positive going the inverting input is at the higher potential and the output of IC1 goes low, and when the input is negative going the inverting input is at the lower potential and the output goes high. It can be seen that the signal is inverted by the trigger circuit, but this is not of practical significance in this application.

The inclusion of R5 and R6 introduces the hysteresis, as these components will shunt R4 when the output is low and thus reduce the bias voltage at the non-inverting input of IC1, but when the output is high they are effectively connected in parallel with R3 and increase this bias voltage. This gives the different negative-to-positive and positive-to-negative switching thresholds. The degree of hysteresis can be adjusted by means of R6, with minimum value corresponding to maximum hysteresis.

Construction

The PCB design for the Cassette Processor unit is shown in **Fig 3.** As the circuit consists of so few components construction should not give any serious difficulties. The CA3140E used in the IC1 position has a MOS input stage and it is therefore advisable to fit it in an IC socket.

A supply voltage of between 5 and 9 volts is required, and the unit can be powered from the 5 volt output at pin 33 of the Oric 1's expansion port. The prototype was however

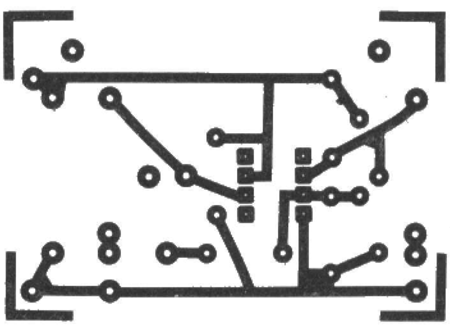


Figure 3. Foil patterns of the conditioners board shown full size.

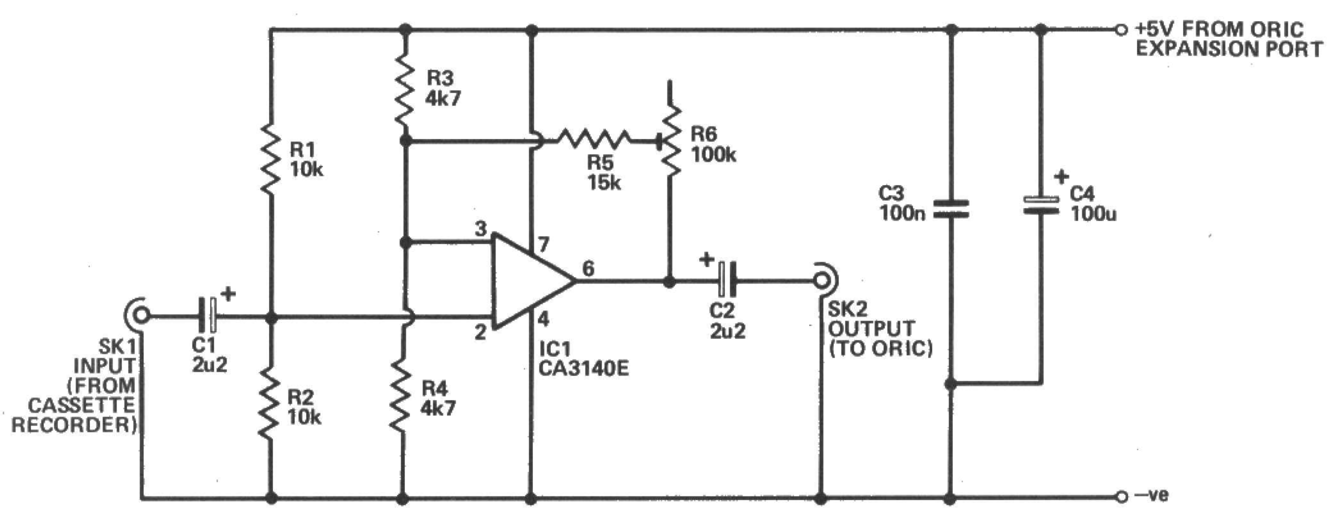
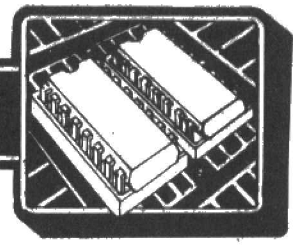


Figure 1. Circuit diagram of the signal conditioner. The circuit consists of a Schmitt trigger circuit with variable hysteresis.



built as a self contained unit powered from a PP3 size 9 volt battery, and fitted in a small plastic box measuring about 120 by 65 by 40mm. If this method is used an on/off switch must be included in the unit.

In use the output to the computer is taken from the output of the processor, and the earphone socket of the recorder is connected to the input of the processor. This will probably require a twin lead fitted with two 3.5mm jack plugs. Initially R6 should be set for about half maximum resistance, and it will only be necessary to try other settings if the unit proves to be ineffective with this initial setting. The adjustment of the volume control on the cassette recorder should not be critical, and it is really just a matter of ensuring that the input level is adequate to reliably operate the trigger. Setting the volume control at or near maximum should be satisfactory.

E&CM

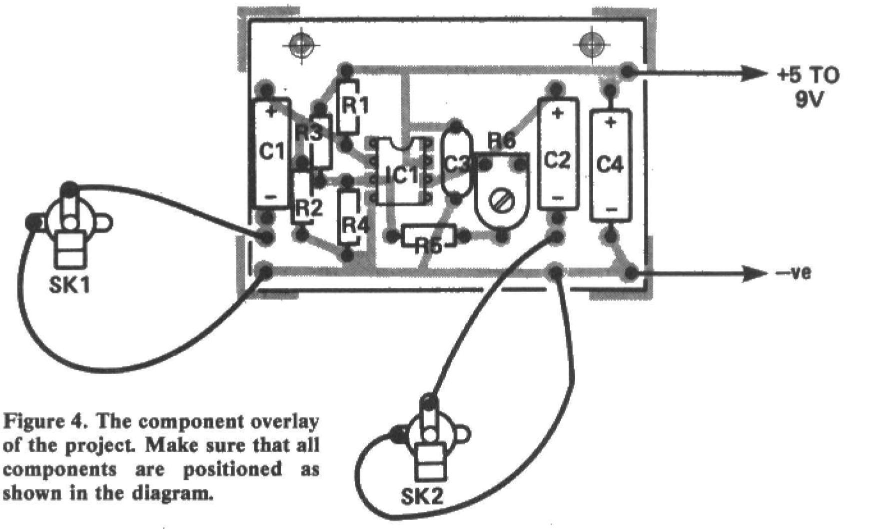
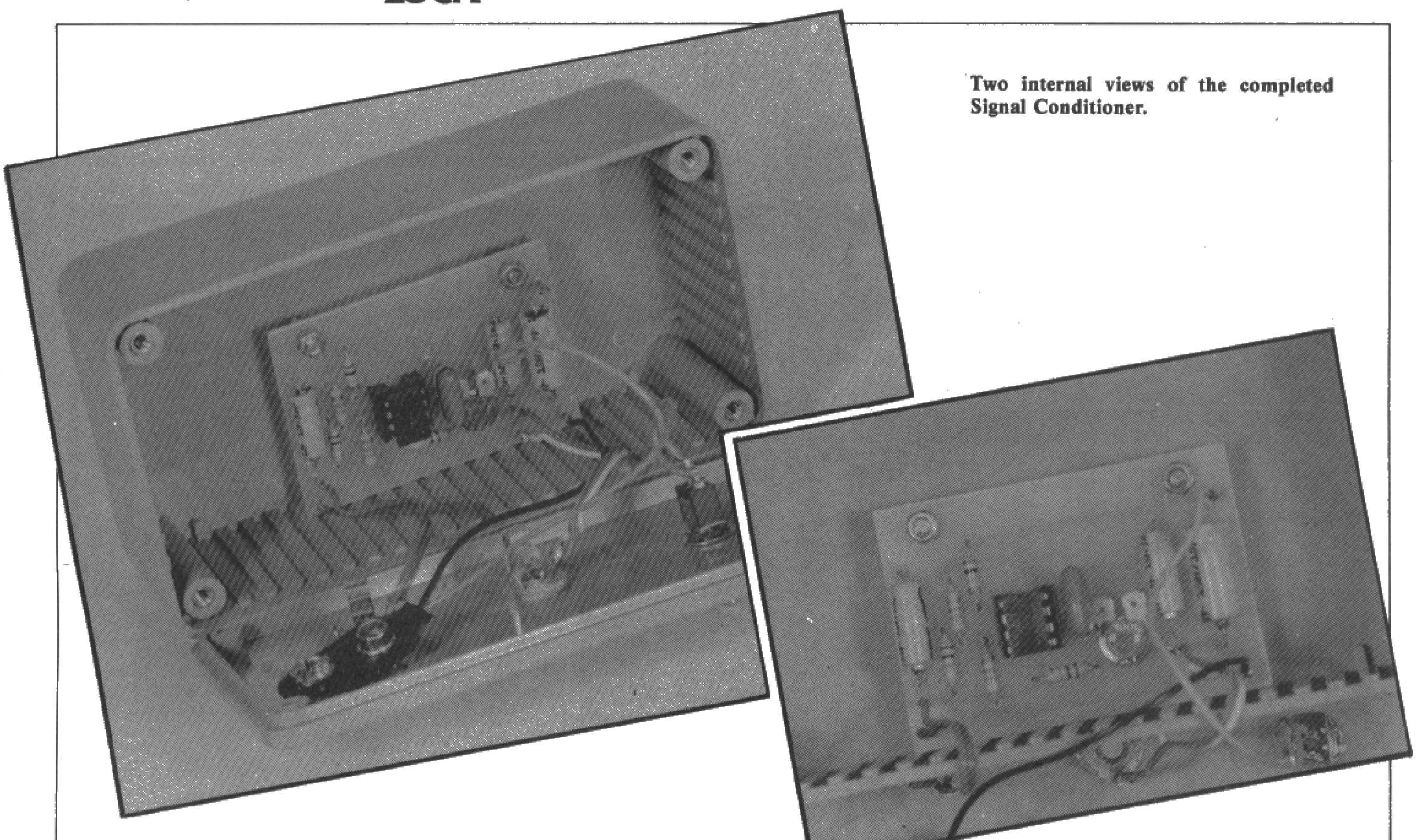


Figure 4. The component overlay of the project. Make sure that all components are positioned as shown in the diagram.



Two internal views of the completed Signal Conditioner.

PARTS LIST

Resistors (all 0.25W 5% except R6)

R1, 2	10k
R3, 4	4k7
R5	15k
R6	100k preset

Capacitors

C1, 2	2u2 63V electrolytic
C3	100nF polyester
C4	100uF 10V electrolytic

Semiconductor

IC1	CA3140E

Miscellaneous

SK1, 2	3.5mm jack socket

Plastic case, DIL socket, PCB, PP3 size battery, battery connector, and on/off switch, or 20 way IDC header socket.

FLEX EXPLAINED

Part 2

Paul Izod and Alan Stirling continue their look at implementing the FLEX OS on the *E&CM* hi-res computer system.

There are two groups of machine code routines that must be added to FLEX before it will run on the target machine. The disk driver routines, which control the operation of the disk drives, and the I/O or console driver routines, which enable FLEX to access a serial terminal, or keyboard and output screen. This second group of routines normally just link FLEX into similar routines already provided as part of your monitor program. Sample routines are provided with FLEX, together with specifications and guide-lines on their operation.

These two routines should be written and checked carefully. At this stage a small program is typed into the machine, which will simulate the operation of FLEX, using the drivers that you have just entered. Using this program, you are able to test the operation of your disk and I/O drivers, by reading and writing sectors to and from the disk, the contents being displayed on your screen or terminal. Since this program which is provided amongst the FLEX documentation, simulates exactly the way in which FLEX will use user routines, once it is operating as specified, the driver routines should be saved onto cassette, rady to be re-loaded once FLEX has been read into the machine for the first time.

This is done by entering another short program (listing also provided) which uses part of the 'Read Sector' routine that you have already tested. This program reads FLEX from the disk into memory. Once FLEX is loaded, reload the disk and I/O routines from cassette. These are designed to overwrite specific parts of FLEX, thus customising it for a particular machine. The acid test occurs when using the system monitor, a jump is made to the 'Cold Start' address of FLEX, to see if all has been in vain! If all is operating correctly, the machine will ask for the date to be entered and then output the FLEX prompt '+++' Success! Be careful not to hit reset and loose all the good work – however the more composed user at this point uses FLEX to save a complete copy of itself back onto the disk, from where it can now be booted when required.

Before FLEX can be booted however, the boot loader program normally stored in the first sector of the disk must be prepared. This program uses the 'Read Sector' routine already used within the disk drivers. Additional code is provided to allow FLEX to be booted from any part of the system disk. Once entered into the machine the command 'PUTLDR' is used to store the program onto the disk.

The final action is to inform the boot loader program of the actual starting track and sector number of the customised version of FLEX. There is a simple command (called 'LINK') that carries out this task. Once completed, FLEX can be booted when required from this disk.

There are a number of other programs and routines which must also be configured for your system, before the implementation is complete. The most important of these is called 'NEWDISK'. It is with this program that blank disks are formatted so that FLEX can use them. The source code for this program is provided as a text file on the FLEX disk and all that need be added is a 'Write Track' routine. This closely resembles the 'Write Sector' routine that will have already been written and tested. Once this routine has been added to the source provided, using the FLEX editor, the complete text file can be assembled using the FLEX assembler. This file is then added to your system disk, along with the other FLEX command files.

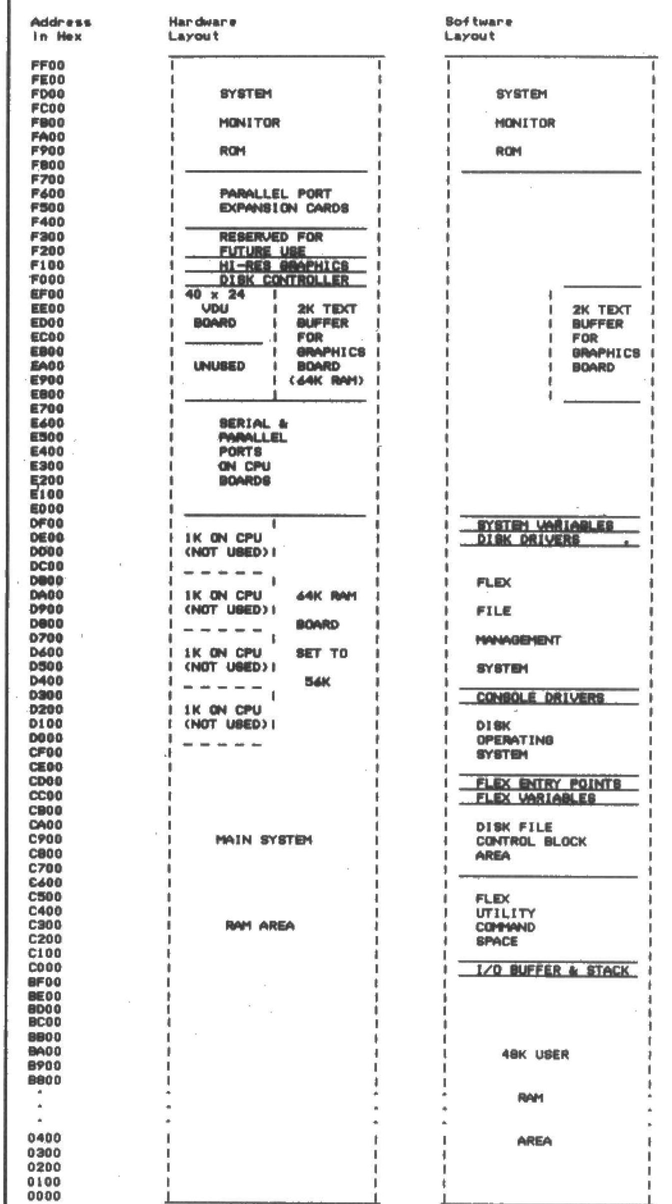


Figure 1. This shows how the *E&CM* hi-res computer is configured to run FLEX. The boards required are the 6809 CPU board, the 64K RAM board and the Disk Controller board. Some means of communicating with the system is also required.

The Contents Of A Flex System Disk

In order to provide more understanding of this operating system, below is listed all the files that are provided on the system disk, together with information as to their use:

Name	Type	Size	Description
FLEX	.COR	22	This is the main 'core' of FLEX, and it is the program to which the dirver routines already discussed, are added.
ERRORS	.SYS	9	A system file, holding a table of error code summaries. This file is automatically searched, if an error occurs.
CAT	.CMD	3	One of the most used commands, this gives a list of all, or selected, files on a disk.
COPY	.CMD	5	Used to copy all, or selected, files from one disk to another, or under a different name onto the same disk.
LIST	.CMD	3	Lists the contents of text files onto the screen. If no drive number is entered, defaults to the working drive.
ASN	.CMD	1	Amends the default drive number for the 'System Drive' or 'Working Drive'. Can be set to 'A11' if required.
DELETE	.CMD	2	Use this command to delete unwanted files. The complete file specification is required.
RENAME	.CMD	1	This will alter the name of a file as it is held in the disk directory.
TTYSET	.CMD	2	A general utility used to set the default operation of your keyboard and screen combination.
P	.CMD	1	Prefixing a command with this letter causes the output to be sent to your printer instead of your screen.
SAVE	.CMD	2	This command enables blocks of memory, normally machine code programs, to be saved onto disk.
EDIT	.CMD	28	A versatile line editor program, which can be used to generate or edit any text file.
ASMB	.CMD	48	The macro assembler provided with the system. The many options allow full control over program assembly.
APPEND	.CMD	3	Generates a single file from the concatenation of a number of others.
BUILD	.CMD	1	A simple text file generator command; quick but without editing facilities.
EXEC	.CMD	1	Allows the machine to be controlled from a text file of commands.
JUMP	.CMD	1	A simple utility to run a machine code program starting from a particular address.
DATE	.CMD	2	Used to alter or display the date as held within the machine.
O	.CMD	2	Redirects output, sending it to a text file on disk instead of to the screen.
LINK	.CMD	1	This program is used to set the boot loader program to the start address of FLEX.SYS, wherever it is on the disk.
VERSION	.CMD	1	Used to display the version number which can be encoded into any program.
PROT	.CMD	1	Enables particular files to be protected from deletion, modification or appearance in disk catalog.
VERIFY	.CMD	1	Controls the 'Read after Write' – Verify, operation of the disk drives.
PRINT	.CMD	2	Used to initiate the printer spooling system – if implemented.
QCHECK	.CMD	4	Controls the files in the print queue waiting for printer spooling.
I	.CMD	1	Commands a program to take input from a text file and not the Keyboard.
XOUT	.CMD	2	This command deletes all files which have been used for printer spooling.
SAVE	.LOW	2	A special version of the 'Save' command, used to save programs occupying the same memory area as 'SAVE.CMD'.
PUTLDR	.CMD	1	Used when first implementing FLEX, to store the boot loader program onto the first sector of the disk.
NEWDISK	.TXT	71	The source code of the 'NEWDISK' program, to be modified with a 'Write Track' routine for the target machine.
LOADER	.TXT	9	The source code of the 'Boot Loader' program, once modified with a 'Read Sector' routine.
PRINTSYS	.TXT	4	The source code of a sample parallel printer driver routine for a 'Centronics' type printer interface.

The following files are not to be found on the FLEX system disk initially, but will have to be provided in order to support a full system. The size of these files may vary from the figures gives, since this will depend upon the amount of extra code added to the original text files.

FLEX	.SYS	25	This is the final operating version of FLEX, combining FLEX.COR, with the new disk and I/O driver routines.
NEWDISK	.CMD	9	This is the assembled version of the 'NEWDISK.TXT' file, with the custom 'Write Track' routine included.
PRINT	.SYS	1	This is the assembled version of the 'PRINTSYS.TXT' file, once modified with the PIA address.
STARTUP	.TXT	1	This file is not mandatory, but if provided, it will operate as an 'EXEC' file on system startup.

This is the basic list of files that make up the FLEX operating system. There is a wide range of extra utilities and programming aids available, all of which can be stored on the system disk as command files and selected when desired.

The FLEX Package

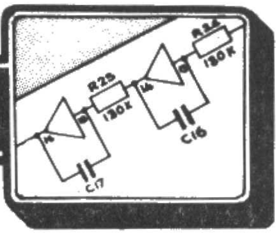
This operating system is available in a form that eases the task of configuring it to run on a non-standard 6809 system. Within the package there are two identical disks, either 5¼" or 8", depending on the size of the disk drives that you wish to use initially. The information provided is identical on either size of disk. Two disks are provided to give you a second chance should you have an accident with one of them! The documentation provided covers the following aspects, in a manual split into 5 parts:

Adaptation Guide – Full instructions and sample listings of programs and routines required to implement FLEX on a 6809 machine.

User's Manual – Full description and reference manual for the operation of the available system commands and utilities.

Text Editor – A complete description of the operation of the content-orientated editing system.

Mnemonic Assembler – Covering the operation and control of the extensive Macro-assembler.



Programmer's Manual – Providing the programmer with the information required to use the available system routines and functions.

FLEX And The Hi-Res Graphics Computer

Figure 1 shows how the HI-RES graphics system should be configured to run FLEX. The main boards required are the 6809 CPU board, 64K RAM board and the Disk Controller board. Some means of communicating with the system is also required. The options are either a serial terminal connected to the ACIA on the CPU board using the S-BUG monitor, the HI-RES GRAPHICS board with keyboard and HI-BUG monitor, or the 40 x 24 VDU board with keyboard and SM-BUG monitor. Due to the design of the disk controller board, users with S-BUG will be unable to boot the disk directly. A new version of S-BUG, called SP-BUG is available. This has the correct disk boot routines. The system will be at its optimum performance if the 4MHz crystal on the CPU board is changed to 8MHz, and the links set to select a 2 micro-second cycle time. In fact this modification converts the system to run at 1MHz. Remember that with the crystal frequency doubled, the baud rate generator on the CPU board will also run at twice the normal speed – the cassette interface may not be too happy about this . . .

As already explained, there are two sets of routines required to link FLEX to any particular 6809 system. In the case of the Hi-Res Computer, the simpler of the two are the Console Driver routines. Since all the recommended Monitor programs for this system use an identical table of indirect jump addresses to routines within the monitor, we can use the same machine code routines to interface either S-Bug, SM-Bug, or the HI-Bug graphics monitor. Unfortunately the jump table in FLEX does not allow the use of indirect addressing. Therefore use has been made of a work area within FLEX between the addresses D370 (hex) and D3E5 (hex), the top of the jump table. Within this area indirect jump instructions are located, which will interface correctly with any of the respective monitor routines.

The Console Drivers

A listing of the assembly code required by FLEX to interface to the system monitor programs (Console Drivers) is available from our offices – please enclose a large SAE. In order to implement these routines just enter the hex codes under the column 'Hex', sequentially into memory starting at the hex location shown under the column 'Addr'. Enter either 1, 2 or 4 bytes as shown.

Some of these routines do not require support initially, and these have been pointed at either RTI, RTS, or fixed memory, so as to disable the particular function. These are marked thus '★'. Once entered and checked carefully against the listing, store these routines on cassette.

The Disk Driver Routines

These routines are rather more complicated than those used in the console driver routines, since there are no useful routines provided within the system monitor program that can be linked into. Therefore these routines have to be complete in themselves; making them much longer than the console drivers.

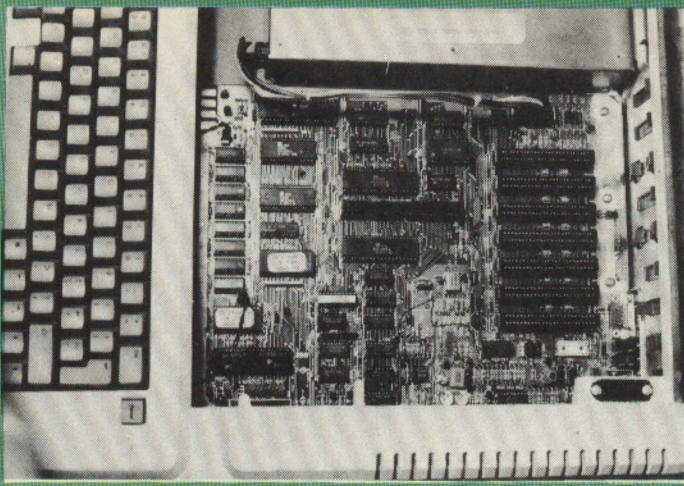
A listing of the disk driver routines is also available from *E&CM's* offices and these have been tested on both 5¼" and 8" drives with FLEX. They only support single sided, single density operation, since this is the easiest way to get the system up and running. Once operating, those extra 'Bells and Whistles' can be added as required. As before, enter the hex codes as listed under the column 'Hex', starting at the hex address locations under the column 'addr'. Once entered check carefully against the listing, and save on cassette.

From this point it is just a matter of following the manual as outlined above, since the two driver routines deal with most of the differences between systems. If other routines are required, these can be developed from those already listed.

Conclusion

Much has been said about the advantages of a user friendly OS on any system. Being a staunch advocate of FLEX, over other, more popular 'Control Programs for Microprocessors', I believe that a FLEX system running on a 56K 6809 system is one of the most powerful single-user systems available today. As one gains experience with such a system, so the true value of this operating system will become apparent.

E&CM



APPLE IIE

Is the Apple IIE a real advance on the pioneering Apple II or a design that lacks the imagination and foresight of the original? Mike James has the answers.

No one can deny that the APPLE II changed the face of computing when it was first introduced. Both this and the Commodore PET are the two machines that brought computers to people who knew nothing of electronics and simply wanted to write programs for, as the advertisements put it, "fun and profit". Before these two revolutionary machines the only low cost computer products were either kits or were not really packaged as standard items of consumer electronics. For the people who accepted these limitations, because they were happy about taking the lid off a computer now and again, and bought one of the early micros, the APPLE II's entry into the market certainly made them wish that they had waited. It's specification was difficult to believe! At a time when other machines needed a VDU to get them going the APPLE II had a built in keyboard and only needed a TV set. While most of the early machines only PROMISED graphics (and black and white at that), the APPLE II had low resolution and high resolution graphics as standard. There are many other features of the APPLE II that made it different for quite some time, but perhaps the most important was that it was the first computer to cram everything into a small typewriter sized case and this was a feature that was obvious to even the least technically experienced. In short the APPLE II managed to capture the public imagination and with it much of the public's money!

It is difficult to say why the APPLE II was so far ahead of the other machines of the period but it is probably related to the fact that it was designed by two engineers in a garage (in the style of so much good equipment of the time). Now APPLE is a large company and the garage days are far behind. The APPLE II's user base is now also large and once people start using a computer they tend to carry on using it and even to encourage others to start using it. This and the large quantity of software available has prolonged the active life of the APPLE II even though there have been better machines around for some time now. As a piece of hardware the APPLE II has to be regarded as history. It is obvious that if APPLE want to remain a large company then they certainly have to produce new products to keep up with the rest of the pack. They have been remarkably slow and cautious about doing this however and while all about them have been proliferating their models, APPLE has introduced only the APPLE III and very recently the super computer LIZA and the APPLE II replacement, the APPLE IIE. Now out of this range of products only the APPLE IIE stands any chance of being of interest to normal mortals and *E&CM* readers! The reason for this is that both the APPLE III and LIZA are aimed at the business market and this is reflected both in price and in hardware flexibility. The APPLE IIE however offers an improved specification of the APPLE II and for less cost. Now, while it is obvious that if you had been planning to buy an APPLE II, because you were already committed to it in some way, then there is no question that you should buy the APPLE IIE in preference; there remains the important question as to whether the APPLE IIE should be considered as a rival to other machines?

What was missing

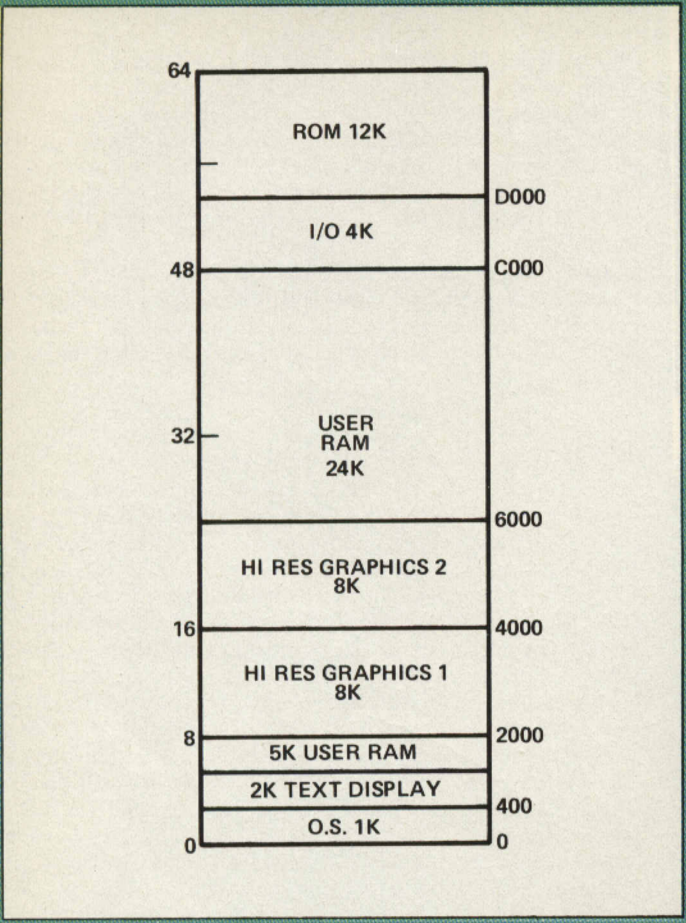
Anyone used to the APPLE II cannot help but notice the major improvement offered by the APPLE IIE – the keyboard. The original II keyboard was very frustrating for its lack of keys that would be considered standard on other machines and because there was no simple way of upgrading it to give lower case ASCII codes. Many small electronics companies have made their reputations (and fortunes) by offering expensive keyboard improvements to APPLE II owners! Now the APPLE IIE solves this serious shortcoming by having a full keyboard capable of producing all 128 ASCII characters along with one or two 'control functions' not included in ASCII. In particular the 'open apple' and 'closed apple' keys familiar to APPLE III users have made the journey down to the APPLE IIE. In this case however these special function keys are connected via the games paddle 'fire button' inputs and this might be something to be kept in mind when writing new APPLE software. All in all the new keyboard is a great relief to any APPLE enthusiast!

To go with the keyboard, the video display has been improved to display both upper and lower case characters as standard. As with the keyboard, this is a definite improvement on the old APPLE. The standard machine is however still limited to a display that is only 40 columns by 24 lines. This can easily be increased to an 80 column display as an extra, but the extra hardware is so minimal that it is difficult to see why APPLE didn't fit it as a standard option. The 80 column card supplied by APPLE fits into a special auxiliary connector placed toward the middle of the main PCB. When in psosition the card looks just like an APPLE II 80 column card in I/O slot 3 so APPLE II software should work fine on an 80 column APPLE IIE. There is also an extended memory card that offers an additional 64K of memory as well as the 80 column display but this,. of course, is more expensive. The firmware to drive either card and much of the extra electronics is already present on the main PCB. As already mentioned the 80 column cards are constructed so as to be compatible with existing APPLE II 80 column cards but in some ways they are better! For one thing both the 40 column and the 80 column video signal come out of the same socket on the back thus obviating the continual socket switching characteristic of so many 80 column APPLE IIs.

Apart from these two great improvements the APPLE IIE offers no major advances over existing APPLE II features. There are however some minor modifications that are steps in the right direction. The games paddles are brough out on the back panel as neat connectors and you no longer need to use a PAL colour card to produce colour displays in the UK but that's about all.

Compatibility

The other features of the APPLE IIE are more by way of extensions and restrictions on the APPLE II's specification rather than putting right omissions in the original design. This said however, it is important to make it clear that although the APPLE IIE is a completely redesigned machine it is as compatible with the APPLE II as is possible following any design change. APPLE tried as many as possible of the existing add on cards, software and other extras ▶



The memory map of the Apple IIE.

while the IIE was still in the design phase. The only devices that won't work are keyboard enhancers, lower case adaptors and memory cards that connect via a flat cable to an IC socket. It is much more difficult to be precise about software difficulties. The new ROMs have the same entry points documented but you may run into trouble if a piece of software uses any internal entry points. Certainly there should be no trouble with any high level language programs – BASIC, PASCAL, PILOT, FORTRAN and LOGO should be fine. Also CP/M software that has been written to use only the BIOS routines should work without modification. The I/O slots at the back of the machine are all present in the APPLE IIE apart from slot 0 which is not great hardship as the original APPLE II slot 0 was very rarely used. Its only real use, to add 16K via a language card, is made redundant on the APPLE IIE by the presence of a full 64K of RAM as standard. As already hinted, slot 3 is in parallel with the new auxiliary connector and this means that I/O cards cannot be used in slot 3 when an 80 column card is present in the auxiliary connector. This is also no problem, because traditionally slot 3 has been used to hold other manufacturers 80 column cards so in general there will be no conflict. An added bonus is the freeing of one extra slot by virtue of the PAL encoder being present on the main PCB. The disk system is still the standard APPLE disk using DOS 3.3 although there are rumours of a new DOS. This is adequate in the sense of being compatible but in no way an improvement. As the machine comes with a full 64K you can run both APPLESOFT and integer BASIC without buying anything extra.

Graphics

The 40 column version of the APPLE IIE offers exactly the same graphics features as the APPLE II. That is, a low-resolution 40 by 40 mode and two lines of text in 16 colours, and a high resolution mode providing 280 by 192 bit-mapped pixels. It has always been difficult to define the number of colours that the APPLE high resolution display can produce because of the odd and rather old fashioned method it uses to produce colour. When the restrictions on where you can plot any particular colour are taken into account, you can achieve 140 by 192 with some restrictions on colour placement or a 140 by 192 unrestricted four colour display. This flexibility would be admirable if APPLESOFT contained colour graphics commands that took the colour restrictions into account but of course it doesn't.

With the 80 column card in place the APPLE IIE can produce double density graphics. Both double-density low resolution and high resolution graphics are possible however, in order to use the deouble-density high resolution feature its also necessary to use the 80 column card with 64K of extra RAM. At the moment there is no software that supports the double-density modes. No doubt software vendors will alter their products given enough time but there will always be a tendency to ignore the extra that the APPLE IIE offers so as to remain compatible with the APPLE II and its larger user base.

The Changing Hardware

The biggest difference between the APPLE II and the APPLE IIE is inside the box where a complete hardware redesign makes things look very different. The original switch mode power supply is still there and the familiar I/O sockets are still to be found along the back of the case but the rest of the main PCB has changed. Although the PCB is about the same size the chip count has gone down considerably. The original APPLE II used around 120 chips but the APPLE IIE provides a slightly better performance with only 31 chips – such is progress! There are two reasons for the reduced chip count. The first is that 64K bit dynamic RAM chips are used to provide 64K bytes using only eight chips. The second is that APPLE (following in the footsteps of UK based Sinclair and Acorn?!) have used two custom-designed chips to handle I/O and memory management.

The overall design is good and solid but it is hardly revolutionary. The CPU is a 6502 as you would expect for an APPLE II compatible machine but it is a 1MHz device and so the APPLE IIE offers no speed improvements. Even though a lot of space has been saved the main PCB's size has not bee reduced. Some of this newly freed area has been used to implement electronics that would normally be extra – the PAL colour encoder for example – but most of it is still unused., At a guess, I would say that the size of the board could have been halved with a consequent saving in cost that would have made the IIE a much more attractive proposition.

Conclusion

All-in-all this is a very conservative redesign. It achieves its purpose of being APPLE II compatible but it fails to make any significant hardware improvements. Some might feel kindly toward APPLE and so claim that for a machine to be compatible it cannot offer wildly new design features, however it only takes a few examples to demonstrate that compatibility is not as restricting as the APPLE IIE makes it seem. For example, one of the big problems with the APPLE II is its crazy memory map with high resolution screens in the middle of user RAM. This problem is corrected, a CP/M Z80 card is used as the user memory is then 're-mapped' to more sensible positions. In fact one of the great pleasures of using the APPLE with CP/M is the simplification of the memory map. The APPLE IIE could also have avoided this messy memory problem by including a variable memory mapping facility, allowing software or hardware re-configuration of memory as required. There would be no compatibility problems with using a switchable 2MHz/1MHz 6502 CPU and so offering a double speed APPLE. There would be no compatibility problems with improving the sound generator to include a sound effects chip of some sort instead of the crude pulse driven speaker. There would be no compatibility problems with placing the character set in RAM so that character definitions could be altered. In fact almost any extra feature could be added as long as it could be disabled at the first sign that is might prove a problem!

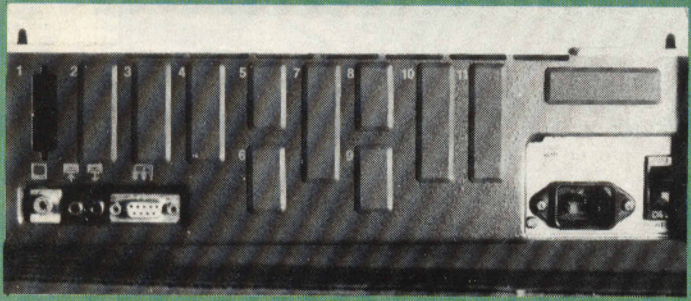
Back to a garage?

The APPLE IIE succeeds in producing a compatible redesigned APPLE II that must be easier and cheaper to manufacture. The cost savings though haven't really been passed on to the consumer. You

can find the IIE on sale at just under £600 but this is not cheap for a machine with its specification. The original APPLE became known as the 'kit machine' because people bought their APPLE and then spend even more money buying the extra cards that were needed to bring the system up to a reasonable standard. Thus the I/O slots that started as a feature in the machines favour rapidly became a liability. The APPLE IIE certainly raises the specification of the standard machine but APPLE's 'buy it as an extra' mentality is still evident in the way the 80 column card is optional.

The APPLE IIE is a better buy than the APPLE II but it is certainly not worthy of the two engineers whose imagination and foresight produced the APPLE II. The new machine is over cautious and fails to offer current APPLE II users a way of achieving a better performance. There are many APPLE II users wondering where to go next and the APPLE IIE is not the way. Quantity users of APPLEs or those that need to use software that isn't available for another machine really have no choice but to use the APPLE IIE. For those that have a free choice however there are more advanced machines, with a better performance, for less money and some of them have been on the market for a few years!

The Apple IIE's keyboard is a distinct improvement on the original and offers the full 128 ASCII character set.



The games paddle connectors are brought out to the rear panel on the IIE. The expansion possibilities are also evident from a glance at this panel.

It seems obvious that the APPLE IIE cannot be the last machine that APPLE offer to the low cost end of the market. The LIZA super computer is a great advance in personal computers but it is expensive and leaves much of APPLE's traditional market uncovered. If APPLE is going to continue to expand its user base then it must offer a new machine that is either low performance and very cheap or high performance at a reasonable price. Perhaps is should provide two of its engineers with a small garage somewhere . . .

E&CM

DRAGON SOUNDS

The Dragon's sound generator has a lot more going for it than might be imagined. Mike James shows, that with a little extra software, the Dragon can be as versatile as any computer in this important area.

The Dragon's sound generator is quite different from the sort of thing you'll find on other micros. Rather than using a special purpose chip with built-in noise and tone generators it uses a D to A converter together with some software to produce a single tone channel. In this sense it can be said that the Dragon's sound is software generated. This is no disadvantage if all that's required is to play simple tunes, but sound effects and chords are a little difficult. The standard BASIC sound commands are well described by the Dragon manual but there are still a few extra pieces of information worth giving. A few novel ways of using the commands are also worth explaining and will be discussed later. To learn about the fundamentals of BASIC Dragon sound however – read the manual first!

The Dragon's Sound Hardware

The complete Dragon sound circuit can be seen in **Fig 1.** The major components are the PIA located at FF20 and the 'one-of-four

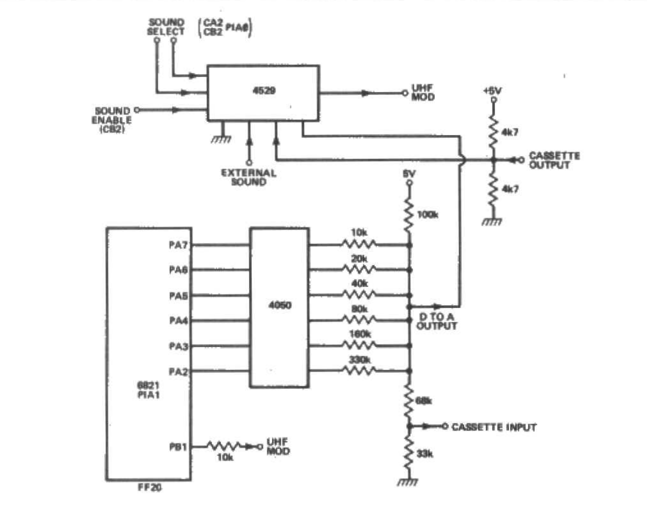


Figure 1. The Dragon's sound circuit

selector' (4529), although the second Dragon PIA at FF00 does play a part in deciding which sound source is heard over the TV's loudspeaker. Six of the PIAs lines, PA2 to PA7, are configured as outputs and then fed to a buffer (4050) and a resistor summing circuit. By a careful choice of the resistor values the outputs from the PIA can be combined to give a voltage proportional to the binary number used to set the output lines high or low. Thus this simple combination of PIA and resistors forms a six bit D to A converter. In other words if a binary number is stored in the PIA's data register, which looks like a fairly normal memory location at FF20, a voltage proportional to that number will be produced at the D to A output. To be more exact if n is the number that corresponds to PA2 to PA7, then the voltage at the D to A output is given by –

$$\text{Voltage} = 0.072n+0.25 \text{ V}$$

Notice that this equation is only to be taken as a guide because the exact value of the voltage produced depends on the accuracy of the resistors and of the five volt power supply to be found in any particular Dragon. The only complication here is that the number that is stored in the PIA's data register is in fact 4*n because the six bits used by the D to A converter start at PA2. PA0 and PA1 are used for other things.

It should be possible to see that by storing a sequence of numbers in the PIA's data register it's possible to produce any particular waveform that is required. This is in fact the way that the Dragon produces most of its sounds. The waveform produced by the SOUND or PLAY commands looks something like that shown in **Fig 2.** This is a very poor approximation to a pure sine wave but for the

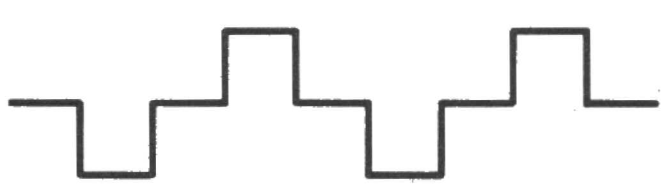


Figure 2. Output from the PLAY and sound commands.

purpose of sound generation it is quite good enough. In fact because pure sine waves sound very boring it is probably better! The D to A converter is also used to generate the two audio tones that are fed to the cassette recorder when saving programs or data. In this case the sine wave is synthesised to a much greater accuracy so that the tape recorder stands a better chance of replaying it without distortion. As a side effect of this dual use the output from the D to A converter is always available at the cassette input. This means that it is possible to record the output from SOUND and PLAY commands simply be setting the cassette to record with the standard tape recorder lead connected.

The D to A converter is the main, but not the only source of sound available to the Dragon. Which one is heard is governed by that which is fed to the UHF modulator and then to the TV set. This selection is carried out by the 4529 'one-of-four' chip in **Fig 1.** Depending on the setting of the 'sound select' and the 'sound enable' inputs to this chip, any one of three possible sound sources can be fed to the UHF modulator. The first sound source is the D to A converter that has already been discussed. The second sound source is the output signal from the cassette recorder and the third sound source is the external sound from the expansion socket. Using BASIC only the first two sound sources can be selected. The D to A is automatically selected by the SOUND and PLAY commands and the cassette source is selected by the AUDIO ON command and de-selected by the AUDIO OFF command.

As well as these three sound sources fed to the 4529 selector there is also a 'single bit' sound source that is connected in parallel with the UHF modulator. This takes the form of a single line (PB1 from the PIA) and so all that it is necessary to do is to 'toggle' this line high and low to form a pulse train. This square wave signal could be of value whenever a rough noise rich in harmonics was required. Normally PB1 is set to act as an input so if an attempt to alter its state by writing to FF22 is made, nothing at all will be heard. It may seem strange that a PIA line described as a sound source is normally set to act as an input. The reason is that its standard use is to detect the start of any signals that have been selected and routed to the UHF modulator. Thus it is either a 'sound source' or a 'sound detector' depending on how you want to think about it.

This is all there is to the Dragon's sound circuitry, no special sound generator chip, just a few standard components. The rest of the sound story is about software.

Pitch And Duration

The simplest sound producing command is –

```
SOUND p,d
```

where p is a number between 1 and 255 specifying the pitch of the tone and d is a number greater than 0 specifying the duration of the tone. This is such a simple command that it is difficult to imagine that there is anything to add to its description! The Dragon manual though doesn't tell you what values of p and d to use to produce any given note for any given time. All that it does tell you is that middle C corresponds to a value of p of 89.

The d parameter in fact specifies the duration of the sound in 4/50ths of a second. This means that the command –

```
SOUND p,s*50/4
```

will produce a sound for s seconds. Working out the effect of the pitch parameter is a little more difficult. The SOUND command could set the pitch of a note either by using interrupts or a delay loop. A quick look at the rest of the Dragon hardware soon reveals that there is no suitable source of interrupts for timing each cycle of an audio note so a delay loop must be used. After some investigation with an oscilloscope, measuring frequencies for various values of p and then fitting the theoretical equation for the time delay produced by going round a loop a given number of times, the following equation was derived:

```
P=256—(167/r)+5*(1—1/r)
```

where r is the ratio of the desired frequency with the frequency of middle C. In other words if Fc is the frequency of middle C then

```
r=F/Fc
```

(Middle C is defined as 261.6Hz on the tempered scale but to use the above equations accurately it is necessary to use the frequency that the Dragon produces for SOUND 89,d.)

As the ratio of notes a semi-tone apart is always 2 (1/12) the above equation can be used to specify notes in terms of the number of semi-tones that they are above or below middle C. For example the following program will play the chromatic scale of C –

```
10 DEF FNN(P)=256—167/2 (P/12)—5*(1/2 (P/12)—1)
20 FOR I=0 TO 12
30 SOUND FNN(I),8
40 NEXT I
```

The function FNN(P) will return the pitch number for any note if P is the number of semi-tones that the note is above or below middle C. This function can be used to produce music outside of the normal range that the Dragon PLAY command allows. In particular it is possible to produce notes one octave above the highest allowed in the PLAY command.

Interrupting The Noise

Any attempt to produce high notes on the Dragon using either SOUND or PLAY will result in a sound that is a little rough and noisy. The reson for this is that the Dragon carries on responding to interrupts every 1/50th of a second to keep TIMER up to date even while a SOUND or PLAY command is being executed. What happens is that just before a TV frame is about to be displayed the 'frame sync pulse' which marks the start of picture is used generate an interrupt which the Dragon's 6809 CPU has to obey. This causes the 6809 to stop what it is doing and jump to the TIMER update routine. This simply adds one to the area of memory used to keep the current time and then allows the 6809 to return to whatever it was doing before it was 'interrupted'. As frame sync pulse occurs every 1/50th of a second this causes the TIMER to be incremented once every 1/50th of a second and this happens (with one or two exceptions) all the time that the Dragon is switched on. This interruption of whatever the 6809 is doing every 1/50th of a second normally makes no noticeable difference because the time taken to update the TIMER is very very small. When it comes to making sounds however, things are very different. For high pitched sounds the regular 1/50th of a second interrupt stops the delay loop that sets the frequency for long enough to make a difference. This regular lengthening of the delay loop causes the rough noise that can be heard at high frequencies. The solution to this problem is easy – simply switch the interrupts off while the sound is being generated. Although this is easy enough to do, the only trouble is that SOUND re-enables the interrupts every time it is executed so defeating our improvements. The PLAY command is quite happy though to produce a tone without initialising the interrupts, except that PLAY uses TIMER to set the duration of the note! The only real solution is to use machine code but to demonstrate the difference between an uninterrupted tone and the standard noisy tone then try the following. First type in –

```
10 TIMER=0
20 PRINT TIMER
30 GOTO 20
```

this will print the current value of TIMER in exactly the way that you might expect. However now add –

```
5 D=PEEK(&HFF03) AND &HFE
6 POKE &HFF03,D
```

and re-run the program you will discover that TIMER is stuck at zero. This is because the two new lines disable the frame sync interrupts and this stops TIMER from being incremented regularly. (Notice that the Dragon is quite good at handling hexadecimal numbers. The function HEX$(n) will return a string that is the hex equivalent of the number n and any number beginning with &H is interpreted as being in hex.) Now type

```
PLAY "O5C"
```

and a very clear tone will be produced but it will not stop automatically. To stop it, it is necessary to press the reset button which also enables the frame sync interrupts again. Following this, once again try PLAY "O5C" and a note of the same pitch will be produced but now with the noise caused by being interrupted 50 times every second. The difference is quite remarkable and it is enough in itself to drive any Dragon owner to machine code!

Moving Sounds

There is another good reason for using Dragon machine code when it comes to sound generation. A common problem to both the SOUND and the PLAY commands is that while any sound is being produced BASIC stops running. This isn't too much of a problem as long as the program is not trying to animate something on the screen while making appropriate sound effects. If this is the case then the best that can be done is to move the object a little, then make some sound and then move the object again and so on. It isn't really possible to do any better than this from BASIC and even using Dragon machine code it is difficult to do much better. Using a faster language like machine code all it is possible to do is to swap between the two actions of making a noise and moving the object so fast that the two things seem to be happening at the same time. This problem of not being able to make a noise while doing something else is a result of the Dragon not using a special sound effects chip that can get on with sound generation all on its own.

PLAYing Sound Effects

The PLAY command is described at great length in the Dragon manual and when it comes to transcribing tunes from sheet music to the Dragon it is difficult to think of anything more convenient. However there is one aspect of the PLAY command that the manual mentions but doesn't really go into enough detail about. The PLAY command can be used to produce simple sound effects as well as pure tones. The principle is to play a string of notes at a very high tempo. For example try –

```
10 PLAY "T155V31O1
   CGCGV20CGCGV10CGCGCGV5CGCGCG"
20 IF INKEY$="" THEN GOTO 20
30 GOTO 10
```

which produces a sound like a gun firing. The PLAY command in line 10 may look a little complicated but it is in fact made up of the same

section repeated. The first part of the PLAY string, "T155V31" sets the tempo and volume to a maximum. The next part "O1CGCG" sets the octave to the lowest possible and then plays the note pair CG twice. This note pair is then repeated at a steadily decreasing volume, V20,V10, and V5 until the end of the sting. The whole effect only works because of the speed with which the notes are sounded. To hear what the play string sounds like normally change T155 to T5 and the effect will vanish!

A wide range of sound effects can be generated by using PLAY in this way, the only trouble is that it takes quite a bit of time to change the play string by trial and error to make the sound that is required. It is helpful to think in terms of a pitch and amplitude graph something like the BBC Micro's ENVELOPE command when trying to invent new sounds.

Selecting The Sound Source

As already mentioned, the Dragon can select one of three different sound sources –

The D to A converter,
The cassette input
and an external sound source.

The way that this selection is carried out is quite simple and although it is really only useful to know how to do it within Dragon machine code it is easier to understand from a BASIC example. The two output lines CA2 and CB2 from the Dragon PIA 0 at FF00 are treated as a two bit binary number which selects the source. Also the output line from PIA 1 at FE20 is used as an enable/disenable bit for the sound output. Putting these details together gives –

```
1000 A=PEEK(&HF001) AND NOT(&H08)
1010 POKE &HF001,A OR (1 AND S)*8

1020 A=PEEK(&HF003) AND NOT(&H08)
1030 POKE &HF003,A OR (2 AND S)*8
1040 A=PEEK(&HFF23) AND NOT(&H08)
1050 POKE &HFF23,A OR (1 AND E)*8
1060 RETURN
```

which will set the sound source to S and either enable it (E=1) or disable it (E=0). The sound source numbers are –

S	source
0	D to A
1	Cassette
2	External sound
3	silence!

Most of the work in subroutine 1000 is in making sure that only the bits that we want to alter are affected by the POKEs. While in BASIC subroutine 1000 is only useful for selecting the external sound output from the expansion socket – pin 35 – and its necessary to use a machine code equivalent of subroutine 1000 when making any sounds without the help of BASIC.

The D/A Converter

It is easy to use the Dragon's D to A converter directly and so in theory at least its possible to produce any sound. In practice however, BASIC is so slow that the best its possible to generate is a low pitched buzz! It is worth examining how BASIC can be used to produce sounds directly mainly so that the ideas can be understood and then translated to machine code but also because it is possible to produce different waveforms even from BASIC.

Using the D to A converter directly is simply a matter of selecting and enabling it using subroutine 1000 given earlier and then POKEing values into the PIA's data register to produce the desired waveform. For example to produce a periodic 'ramp' waveform use –

```
10 S=0
20 E=1
30 GOSUB 1000
40 FOR V=0 TO 63 STEP 4
50 GOSUB 2000
60 NEXT V
50 GOTO 40
```

```
2000 POKE &HFF20,V*4
2010 RETURN
```

Subroutine 2000 will store the contents of V in the PIA data register starting at PA1. To produce a square wave try –

```
10 S=0
20 E=1
30 GOSUB 1000
V=0
50 GOSUB 2000
60 V=63
70 GOSUB 2000
80 GOTO 40
```

It will soon be discovered that from BASIC the highest frequency that can be produced is very low! It is important though to realise that this is entirely due to BASIC's lack of speed. If the same technique were to be used in machine code then its really possible to gain control of the Dragon's sound.

Single Bit Sound

There is one other sound source that the Dragon can use – single bit sound. This is simply a PIA line that is connected directly to the UHF modulator so there is no need to select or enable it. However as it is normally set to act as in input it is necessary to first change the line to act as an output. Once it is an output then all we can do is to change it from 0 to 1 and vice versa so producing a square wave. For example –

```
10 GOSUB 1000
20 B=0
30 GOSUB 2000
40 B=1
50 GOSUB 2000
60 GOTO 20

1000 A=PEEK(&HFF23) AND NOT(4)
1010 POKE &HFF23,A
1020 A=PEEK(&HFF22) OR 2
1030 POKE &HFF22,A
1040 A=PEEK(&HFF23) OR 4
1050 POKE &HFF23,A
1060 RETURN

2000 A=PEEK(&HFF22) AND NOT(2)
2010 POKE &HFF22,A OR 2*B
2020 RETURN
```

Subroutine 1000 sets PB1 to act as an output and subroutine 2000 is then used to set it high and low alternately. Once again the comment about not being able to produce high frequencies from BASIC holds.

AUDIO and MOTOR

The use of the two commands AUDIO and MOTOR is well explained by the Dragon manual. The command AUDIO ON will connect the output of the tape recorder to the UHF modulator and AUDIO OFF will disconnect it. The recorder's motor can be stopped and started using MOTOR ON and MOTOR OFF but only if the remote control connection is being used to gain familiarity with these two commands place a music cassette in the tape recorder, press the play button and then try –

```
10 PRINT "PRESS 'H' TO HEAR"
20 PRINT "PRESS 'S' TO STOP"
30 MOTOR ON
40 A$=INKEY$
50 IF A$="H" THEN AUDIO ON
60 IF A$="S" THEN AUDIO OFF
70 GOTO 40
```

The possible uses for the AUDIO command include things like recording a sound track of questions for a quiz program or giving instructions in spoken English etc. In fact as the Dragon has the AUDIO and MOTOR commands who needs a speech synthesiser!

The only trouble is synchronisation. How is it possible to make sure that the program keeps in time with the tape? The answer is to use the sound sense input that was described in the last section as the single bit sound output!

In its standard role as an input this PIA line can be used to detect the presence of an AUDIO signal playing over the TV speaker. The following function will return a 0 if the speaker is silent and a 1 if there is any sound –

```
10 DEF FNS(X)=1–(PEEK(&HFF22) AND 2)/2
```

(Notice that X is a dummy variable and is only included because the Dragon will give an error without it). Using this function a program can be synchronised to a tape by leaving quiet sections before the start of each 'chunk' of sound. For example find a blank cassette and record the words 'ONE', 'TWO', 'THREE' on it with a good silent gap between each. It is important that the silences between each word are very quiet and each work is clear and loud. Then rewind the tape and set it to play. If the single bit sound program given in the last section was tried, switch the Dragon off and then on again to reset the single bit sound line to input. Now try –

```
10 DEF FNS(X)=1–(PEEK(&HFF22) AND 2)/2
20 MOTOR ON
30 AUDIO ON
40 IF FNS(X)=0 THEN GOTO 40
50 PRINT "ONE"
60 IF FNS(X)=1 THEN GOTO 60
70 IF FNS(X)=0 THEN GOTO 70
80 PRINT "TWO"
90 IF FNS(X)=1 THEN GOTO 90
100 IF FNS(X)=0 THEN GOTO 100
110 PRINT "THREE"
120 FOR I=1 TO 100
130 NEXT I
140 MOTOR OFF
```

By adjusting the volume control on the tape recorder it should be possible to get the words ONE TWO THREE to appear on the screen. In practice it's necessary to be careful not to trigger programs on the wrong sound. Any problems in this area are almost certainly due to the fact that the silences between the words are not quiet enough. This is only a demonstration program but it can be seen that this same method could be used to text for the start of any recorded message on tape.

Beyond BASIC

The Dragon's sound capabilities are greater than the BASIC commands SOUND and PLAY would lead one to believe. Using machine code it is possible to produce white noise, chords, sound effects and even computer generated speech! The Dragon's simple sound hardware makes it very flexible but to make use of this it is necessary to have the right software, and BASIC is just too slow. If you decide to stay with BASIC however, then there is still a lot that can be achieved by understanding the hardware and experimenting.

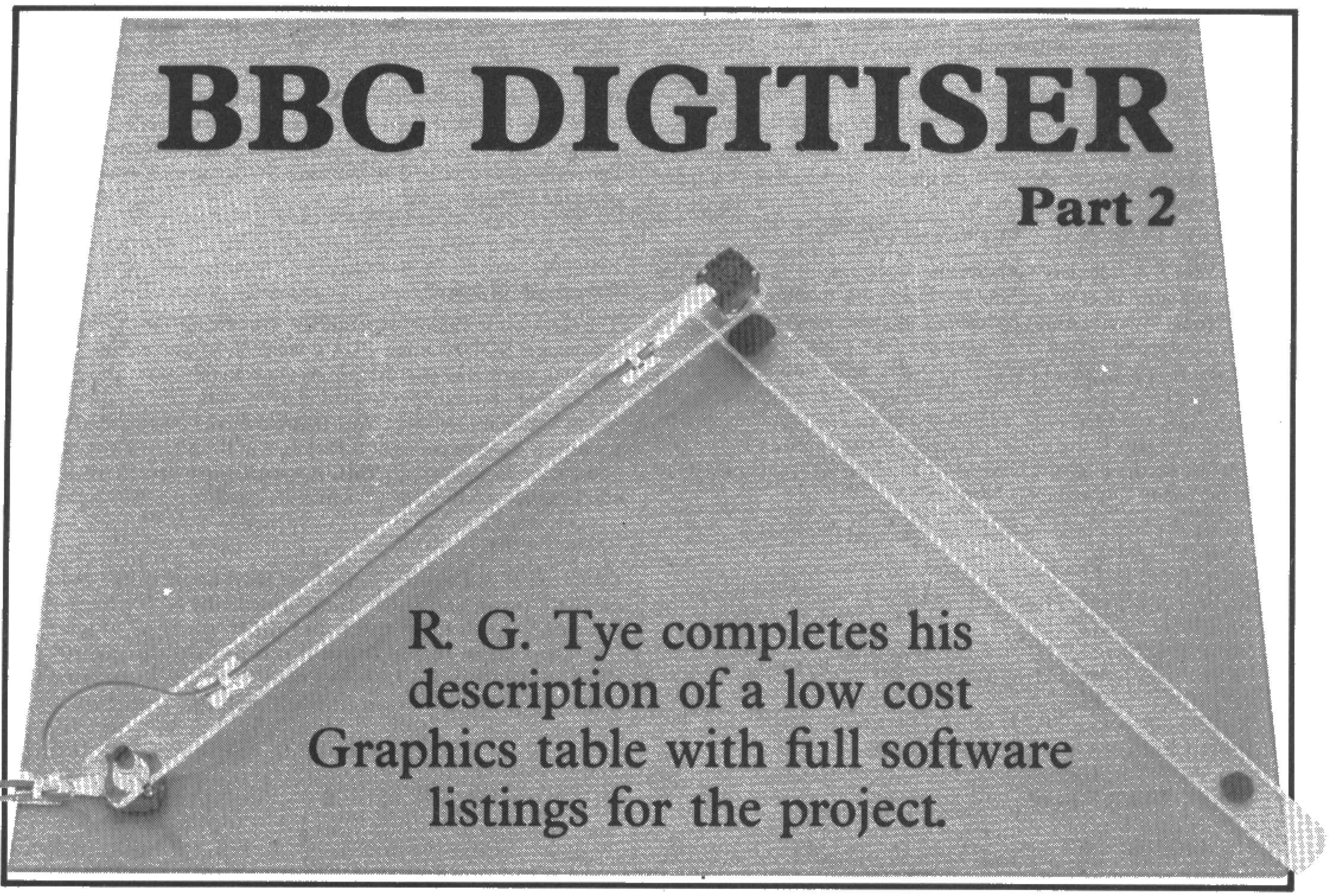
For those of you wishing to find out more about sound generation on the Dragon and about Dragon machine code programming, the author has written two books which might be of interest.

"The anatomy of the Dragon" is to be published shortly while "The Language of the Dragon – 6809 assembly programming" is due to be published in the Autumn. Both are from Sigma Press.

E&CM

BBC DIGITISER

Part 2



R. G. Tye completes his description of a low cost Graphics table with full software listings for the project.

The movements of the 'tracing pointer' on the articulated arm may be reproduced on the screen either as a line drawing or as blocks of colour. Information is displayed on the screen to help you.

These displays will be explained in the notes that follow. The program permits you to select different MODEs and expand or contract the drawing by entering expansion factors before you begin the graphics. (Try X factor 2.0, Y factor 0.5). It is also possible to opt for the X and Y co-ordinates of the tracing point to be displayed at the top of the screen. Instructions are given on the 'adjustment page' at the beginning.

All graphics may be saved as a tape recording which is easy to merge with another routine, so that this program may be used to prepare graphics data for storage and use in subsequent programs.

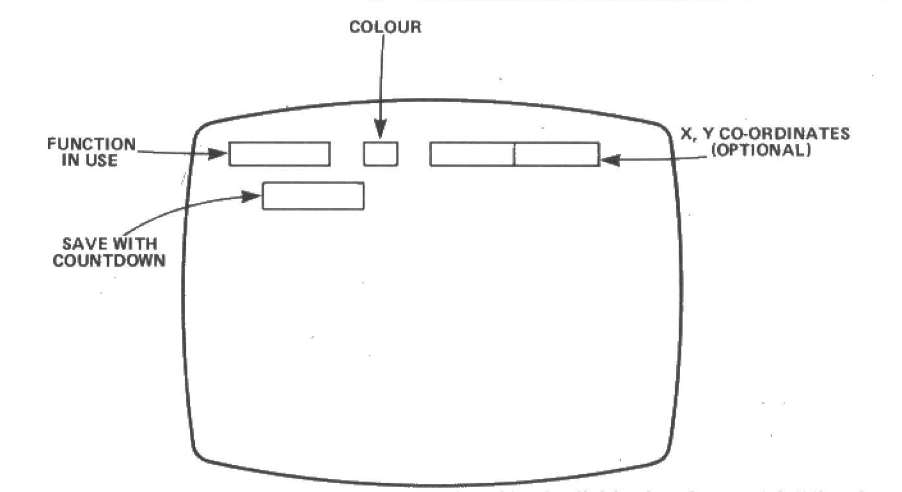


Figure 1. A representation of the display produced by the digitiser's software. A full list of functions are shown in Table 1.

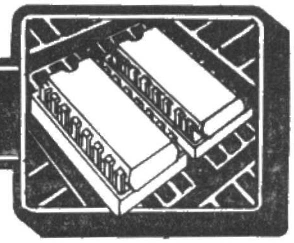
PROGRAM 1

```
 10 MODE 7:VDU 23;8202;0;0;0
 20 REM GRAPHICS TABLE SET-UP
 30 REM EQUAL LENGTH ARMS ONLY
 40 M=9.738:F=6.234
 50 INPUT"ENTER YOUR MAGIC NUMBER"MAGIC
 60 MT=INT(MAGIC/M)
 70 FT=INT(MAGIC/F)
 80 PRINTTAB(0,15) "TARGET-moving pot.";MT
 90 PRINTTAB(0,20)"TARGET-fixed pot.";FT
100 PRINTTAB(28,15) ADVAL(1)DIV10
110 PRINTTAB(28,20) ADVAL(2)DIV10
120 FOR delay=1TO200:NEXT
130 PRINTTAB(28,15)"      "TAB(28,20)"
140 GOTO 100
```

Program Structure

The main program is essentially linear in construction, consisting of a loop which includes the calculations, keyboard interrogation for function selection, and an executive section, which draws on the screen. The length of this loop and the deliberate non-disabling on the two unused ADVAL channels is used as a time delay. When the 'Save' function is in use, the addition of this procedure into the system brings the total delay to an acceptable figure to ensure that your time does not run out before the graphics can be completed and stored.

The data is stored as numbers which are in order X co-ordinate., Y co-ordinate, colour, function. No attempt has been made to economise on memory, these numbers being simply written into four arrays. This, and the fact that the entire program is written in BASIC, ensures that DATA is freely accessible for modification by inexperienced



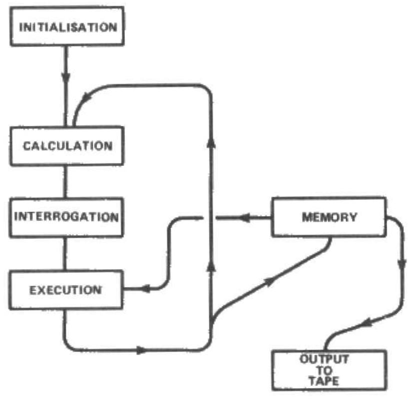


Figure 2. A flow chart representation of the listing shown as Program 2.

PROGRAM 3

```
   10INPUT"ENTER REPLAY MODE    "
mode
   20MODEmode
   30PROCDRR
   40END
  999:
 1000DEF PROCDRR:REM  (C) R.G.TY
E
 1010REM DrawReproductionRoutine
 23JAN83
 1020VDU23,234,255,255,255,255,2
55,255,255,255
 1030VDU23,235,255,129,129,129,1
29,129,129,255
 1040:
 1050REPEAT
 1060READXD%,YD%,Y%,Q%
 1070IFQ%=99 OR Q%=0 GOTO 1150
 1080 GCOL0,Y%
 1090REM=* Spare line *=
 1100ON Q% GOTO 1150,1110,1120,1
130,1140,1150
 1110MOVEXD%,YD%:GOTO1150
 1120DRAWXD%,YD% :GOTO1150
 1130PROCtint :GOTO1150
 1140VDU5:GCOL0,1:MOVEXD%,YD%:PR
INTCHR$235:GCOL0,0:MOVEXD%,YD%:P
RINTCHR$234:VDU4
 1150UNTIL Q%=99
 1160:ENDPROC
 1170DEF PROCtint :dY%=4
 1180XX%=XD%:XZ%=XD%:YY%=YD%

 1190REPEAT:REPEAT
 1200PZ%=POINT(XZ%+4,YY%)
 1210PX%=POINT(XX%-4,YY%)
 1220IFPX%=0 PLOT69,XX%,YY%:XX%=
XX%-4
 1230IFPZ%=0 PLOT69,XZ%,YY%:XZ%=
XZ%+4
 1250UNTILPX%<>0 AND PZ%<>0
 1253XX%=(XX%+XZ%)DIV2:XZ%=XX%

 1255YY%=YY%+dY%:UNTILPOINT(XX%,
YY%)>0
 1260IFdY%=-4ENDPROC ELSEdY%=-4:
GOTO1180
```

programmers. The more erudite among you can easily derive systems for saving memory if you need it. (Note – if you have the disc interface you will need to enter PAGE=&E00 before loading this program).

TABLE 1

M – move	The arm may be moved anywhere without affecting the screen. A small blinking cursor shows the position of the tracing point.
D – draw	Lines are drawn across the screen as traced out by the arm.
J – join	Straight lines between points may be drawn by marking a point with D, then pressing J and moving to a second point. Press D again, and the points will be joined. Most useful for copying line diagrams.
E – erase	A blinking cursor appears on the screen which may be guided by the arm. It erases all it touches.
C – clear	Clears the screen completely.
T – tint	Position the Move cursor inside an enclosed figure and press T, the area will be filled with colour. Very complicated shapes require several "tints" to fill them completely. This function also prints the relative areas filled upon the screen (In the function space). If you operate T consecutively the areas are added, but if you use any other function in betweėn (e.g. M) the area value is set to zero. Be sure your figure is totally enclosed or the colour will leak!
P – print	The number and lower case letters, but not the "Initials" may be printed at any point on the screen. Use the blinking Move cursor to position (it indicates the upper left hand corner of the character), press P, then the character you wish to print. This function cannot be saved in the memory.
S – save	Puts all the screen data into the computer's memory (except P). 250 points may be stored in MODES 1 and 2, and 500 in MODEs 4 and 5. A countdown (with 'Bleeps') is provided to help you. If the memory is full you will get a Reset instruction.
F – finished save	Used when sufficient points have been saved and cancels that function. Automatic if all the memory is used up.
G – get from memory	Recalls the memory and redraws the graphics upon the screen. Cannot be used until Save is Finished. Clear the screen or you won't see the effect.
R – reset	Cancels everything and starts again.
O – open to tape	Prepares to tape record the data in the memory so that you may use it in another program. Instructions are provided. Be sure your cassette recorder is ready!

Functions

The main program (CHAIN "DRAW") can now be used to draw some graphics. Functions are selected by pressing the key for the appropriate initial letters. Colours are selected (according to MODE) by pressing numbers 1-9. Try 1 to start. A full list of options is shown in **Table 1.**

Note that DATA is recorded as an ASCII code on the tape, and cannot be LOADed in the normal way.

Program 3 is a PROCedure which may be used in programs to reproduce the graphics stored on tape. To use this facility, type in DEFPROCDRR and DEFPROCtint (from the main program) and then merge your data recording by typing *EXEC"" (and RETURN). As a graphics tape is played, the DATA lines will be automatically entered as though they were typed in, so ensure that the line numbers do not clash.

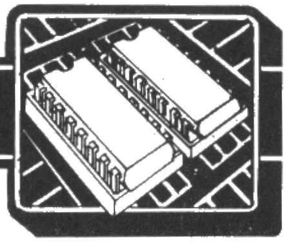
Pauses in the tape recorder operation are normal with these operations, and do not mean it has crashed!

The DATA may be mathematically manipulated by inserting instructions, for example, adding 600 to each X co-ordinate will ensure that your graphics only appear in the right hand side of the screen! It is also permissable to 'replay' in a different MODE to that used originally, try it and see.

Operating Systems

This program was written using OS 0.1, but has been checked and found to run on OS 1.2. If you have a disc interface it will be necessary to enter PAGE=&E00.

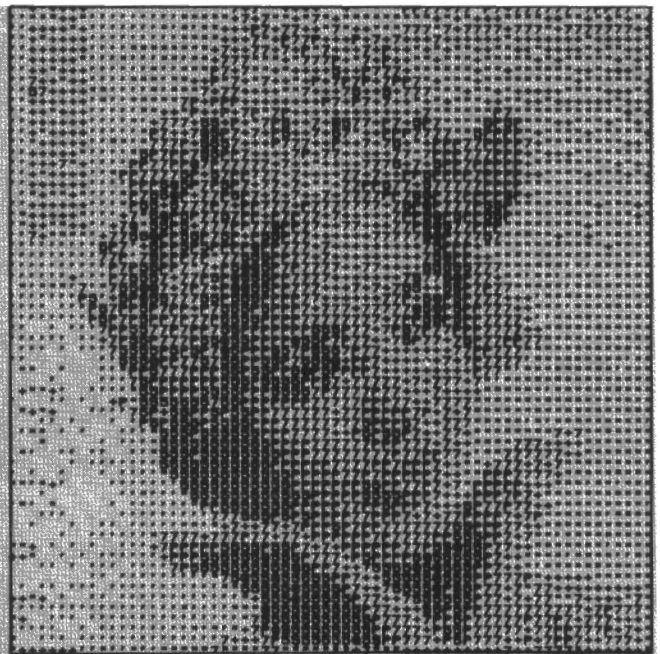
Experimenters with OS 1.2 might like to try using PLOT 77,X,Y as an alternative to the "tint" routine. If you get it to work, please let me know.



PROGRAM 2

```
>LISTLIST
10*KEY0"|M MO.7 |M |N *FX11,50 |M L.|M"
20REM=* GRAPHICS TABLE *= (c) R.G.TYE
30REM "DRAW" 24JAN1983. Articulated arm+linear pot. sensors.
40ONERRORRUN
50MODE7:key=FALSE:store%=2:Q%=2:*FX16,2
60file=FALSE:get=FALSE:coord=TRUE:*FX11,0
70VDU23,234,255,255,255,255,255,255,255,255 :VDU23,235,255,129,129,129,129,129,255
80key=FALSE:store%=2:A%=0:*FX16,2
90PROCheader:PROCadjust
100MODEmode%:PRINTTAB(0,2)"?":VDU23,0,11;0;0;0
110IFmode%<=2 THEN size%=250 ELSE size%=500
120DIM XD%(size%),YD%(size%),Y%(size%),Q%(size%):J%=size%:K%=size%
130VDU 23;8202;0;0;0
140VDU4:IF get=TRUE THEN 390
150THA=ADVAL(1)/15642:REM 240 DEG POT
160THB=ADVAL(2)/15642
170THC=(THA+THB)
180X=COS(THB) :Y=SIN(THB)
190XX=COS(THC) :YY=SIN(THC)
200XD%=INT((X-XX)*1000*FX)-200
210YD%=INT((Y-YY)*1000*FY)
220IF coord=TRUE THENPRINTTAB(9,2)"     " TAB(16,2)"     " :PRINTTAB(8,2)"X";XD% TAB(15,2)"Y";YD%
230Q$=INKEY$(0):IF Q$="" THEN250
240IFQ$<>"T"THEN A%=0
250IFQ$="R"THEN RUN
260IFQ$="S"THENPRINTTAB(0,3)"SAVE":file=TRUE:key=TRUE
270IFQ$="F"THENPRINTTAB(0,3)"    ":file=FALSE
280IFQ$="O"AND file=FALSE AND key=TRUE THEN840
290IFQ$="J"THEN Q%=1:PRINTTAB(0,2)"Join "
300IFQ$="M"THEN Q%=2:PRINTTAB(0,2)"Move "
310IFQ$="D"AND Q%>=5THEN Q%=2::PRINTTAB(0,2)"Move ":GOTO330
320IFQ$="D"THEN Q%=3:PRINTTAB(0,2)"Draw "
330IFQ$="T"THEN Q%=4:PRINTTAB(0,2)"Tint "
340IFQ$="E"THEN Q%=5:PRINTTAB(0,2)"Erase"
350IFQ$="C"THEN CLG:Q%=2:file=FALSE:PRINTTAB(0,2)"?"
360IFQ$="G"AND file=FALSE AND key=TRUE THEN get=TRUE:PRINTTAB(0,2)"GET ":K%=size%
370IFQ$="P"THEN Q%=6:PRINTTAB(0,2)"Print"
380IFVAL(Q$)<1ORVAL(Q$)>9THEN390ELSEIFQ%<6THENY%=VAL(Q$):COLOURY%:PRINTTAB(6,2);Y%
390IF Q%=1 THEN 140
400IF file=TRUE THEN GCOL0,Y%:PROCfile
410IF get=TRUE THENPROCget
420 GCOL0,Y%
430IFQ%=2 PLOT70,XD%,YD%:FORd=1TO40:NEXT:PLOT70,XD%,YD%:GOTO140
440IFQ%=3 DRAWXD%,YD%:GOTO140
450IFQ%=4 PROCtint:Q%=2:PRINTTAB(0,2);A%:IFfile=TRUE J%=J%-1:GOTO140
460IFQ%=5 :VDU5:GCOL0,1:MOVEXD%,YD%:VDU 235:GCOL0,0:MOVEXD%,YD%:VDU 234:GOTO140
470IFQ%=6 VDU5:IF Q$="P" THEN 140 ELSEMOVE XD%,YD%:PRINTQ$ :GOTO 140
480DEF PROCheader
490 CLS :FORI=1 TO 2
500PRINTCHR$157TAB(10)CHR$141CHR$132"GRAPHICS TABLE"
510NEXT:ENDPROC
520DEF PROCadjust
530 mode%=5:FX=1:FY=1
540PRINT'CHR$134"ARE YOU PLUGGED IN ??  KEY REMINDERS."
550PRINT'CHR$134"Move  Draw  Erase  Join  Clearscreen"
560PRINTCHR$134" Save  Print  Tint  Re-set(and ESCAPE)"
570PRINTCHR$134"  Finishedsave _ Getmemory  Opentape"
580PRINTCHR$134"Initals for functions, numbers colour."
590PRINT'"User adjustable factors:_"
600PRINT'"Default values are shown in brackets,  to accept"CHR$130"ALL"CHR$135"of these now you need onlypress"CHR$130"SPACE,"CHR$135"else press"CHR$131"ANY letter,"
610PRINT"except"CHR$129"E for EXIT."
620PRINT'CHR$(131)"MODE 1,2_250; 4,5_500 pts. (5)"
630PRINT'CHR$(131)"EXPANSION FACTOR X (1.0)"
640PRINT'CHR$(131)"EXPANSION FACTOR Y (1.0)"
650PRINT'CHR$(131)"DISPLAY CO-ORDINATES Y/N (Y)"
660G$=GET$:IFG$=" " ENDPROC ELSE IFG$="E" GOTO1050
670 INPUTTAB(32,17),mode% TAB(32,19),FX TAB(32,21),FY TAB(32,23),coord$
680IF coord$<>"Y" THEN coord=FALSE
690ENDPROC
700DEF PROCfile
710IF J%=0 THEN PRINTTAB(0;2)"RESET":GOTO760
720IF Q%=3 AND store%=2 THEN QQ%=2 ELSE QQ%=Q%
730XD%(J%)=XD%:YD%(J%)=YD%:Y%(J%)=Y%:Q%(J%)=QQ%
740 store%=Q%
750IFQ%=2 ORQ%=6 ENDPROC ELSE J%=J%-1:PRINTTAB(5,3);J%
760IFJ% MOD50=0 OR J%=10THEN PRINTCHR$7 :PRINTTAB(5,3)"    "
770IF J%=0 THEN file=FALSE:PRINTTAB(1,3)"    "
780ENDPROC
790DEFPROCget
800XD%=XD%(K%):YD%=YD%(K%):Y%=Y%(K%):Q%=Q%(K%)
810K%=K%-1
820IF K%=J% THEN Q%=2:get=FALSE:PRINTTAB(0,2)"?      "
830ENDPROC
840MODE7:PROCheader
850PRINT'CHR$131"ARE YOU READY TO RECORD ?"'
860PRINTCHR$130"Press"CHR$131"Y"CHR$130"to carry on _ any letter to "
870PRINTCHR$130"return to main program."
880IFGET$<>"Y"THEN MODEmode%:PRINTTAB(0,2)"?":Q%=2:GOTO140
890INPUT'"Please name your file ",D$
900IFD$="" THEN890
910INPUT'"Starting line number ",N%:PRINT
920D=OPENOUT(D$)
930K%=size%+2:REPEAT
940K%=K%-2:IFK%<=J% THEN L$=STR$(N%)+"DATA0,0,0,99":GOTO960
950L$=STR$(N%)+"DATA"+STR$(XD%(K%))+","+STR$(YD%(K%))+","+STR$(Y%(K%))+","+STR$(Q%(K%))+","+STR$(XD%(K%-1))+","+STR$(YD%(K%-1))+","+STR$(Y%(K%-1))+","+STR$(Q%(K%-1))
960FOR I=1TOLEN(L$)
970BPUT#D,ASC(MID$(L$,I,1))
980NEXT
990BPUT#D,13
1000 N%=N%+2:UNTIL K%<=J%
1010CLOSE#D
1020PRINT'CHR$134 D$;" IS SAVED."
1030PRINT'CHR$134"USE *EXEC TO MERGE DATA."CHR$136" BACKUP? Y/N"
1040IF GET$="Y" CLS:PROCheader:GOTO890
1050*FX16,4
1060*FX11,50
1070CLS:PRINT'"(C) R.G.TYE, 'BYE":END
1080DEFPROCtint: dY%=4
1090XX%=XD%:XZ%=XD%:YY%=YD%
1100REPEAT:REPEAT
1110PZ%=POINT(XZ%+4,YY%)
1120PX%=POINT(XX%-4,YY%)
1130IFPX%=0 PLOT69,XX%,YY%:XX%=XX%-4:A%=A%+1
1140IFPZ%=0PLOT69,XZ%,YY%:XZ%=XZ%+4:A%=A%+1
1150UNTILPX%<>0 AND PZ%<>0
1160XX%=(XX%+XZ%)DIV2:XZ%=XX%
1170YY%=YY%+dY%:UNTIL POINT(XX%,YY%)>0
1180IFdY%=-4 ENDPROC ELSEdY%=-4:GOTO1090
```

Picture 1. This display is built up using standard letters with overprinting for the four darkest levels.



Picture 2. Produced with an Epson MX80 in subscript mode. This limits the grey scale to eight levels.

Computer Picture Printing

C. Grant Dixon (G8CGK) follows up our recent article on the Spectrum Image Processor with a different technique of computer image printing.

The three pictures which accompany this article were loaded into computer memory using a type of digital scan-converter which accepts fast-scan TV and delivers a 4-bit slow-scan signal which can be read at an input port and stored in computer memory. **Pictures 1** and **2** are 64 x 64 picture elements (pixels) and **Picture 3** is 107 x 108 pixels.

Picture 1 is printed using standard letters with overprinting for the four darkest levels in the 16 step grey-scale. The stored picture grey levels run from 0 to F (hex) and by using a suitable look-up table the characters corresponding to these grey levels are selected. A second pass of the print head using another table overprints the characters as shown in **Table 1.**

Picture 2 is printed by an EPSON MX80 F/T 3 in the subscript mode. Owing to the limited range of densities which can be obtained with the letters available, the printing is restricted to eight levels and no overprinting seems to be necessary.

The characters used are shown in **Table 2.**

For both **Pictures 1** and **2** the characters were chosed by printing a block of the same character, cutting out the blocks as paper squares, and arranging these to give the best grey-scale; the appropriate letters could then be chosen.

Picture 3 is 107 x 128 pixels so as to get the characteristic slow-scan square format. Ideally it should be 128 x 128 pixels, and this is possible with an MX82 printer which uses the same dot spacing vertically and horizontally. Each pixel is represented by a matrix of 4 x 4 dots which allows for 17 grey levels, i.e. By printing 0 to 16 dots in the pixel area as required. Care is needed in the choice of the pattern of dots to avoid an undesirable overall patterning when areas of the picture are at the same grey level. **Figure 1** shows a set of pixels that avoid any patterning problems. A look-up table is used to select the correct dots for the required grey level and this operation is then performed for the pixel in the line below; these two sets of four bits are printed together so that as the print head traverses across the paper it is printing *two* picture lines. The grey scale in the picture has been added by writing a routine to modify the area of memory where the picture is stored.

For those readers with an experimental turn of mind, the stored video data may be further processed by a secondary look-up table prior to printing. For instance, to increase the contrast of the picture we could use the following relationship shown in **Table 3.**

The darker shades of grey have been reduced in value and the lighter shades made even lighter. Complementing the data will give a negative picture and many other effects are possible, limited only by the imagination of the experimenter.

Different dots

Finally it may be of interest to compare the pictures produced by different pixel formats. **Picture 4** shows Ross-On-Wye Market Hall using the pixels as described in *E&CM's* Spectrum Image Processor article (June & July '83) while **Picture 5** shows the same scene produced using the pixel patterns shown in **Fig. 1.**

These two pictures clearly show the importance of the pixel elements that go to make up a computer generated picture.

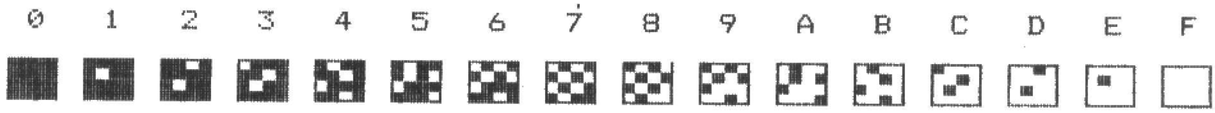
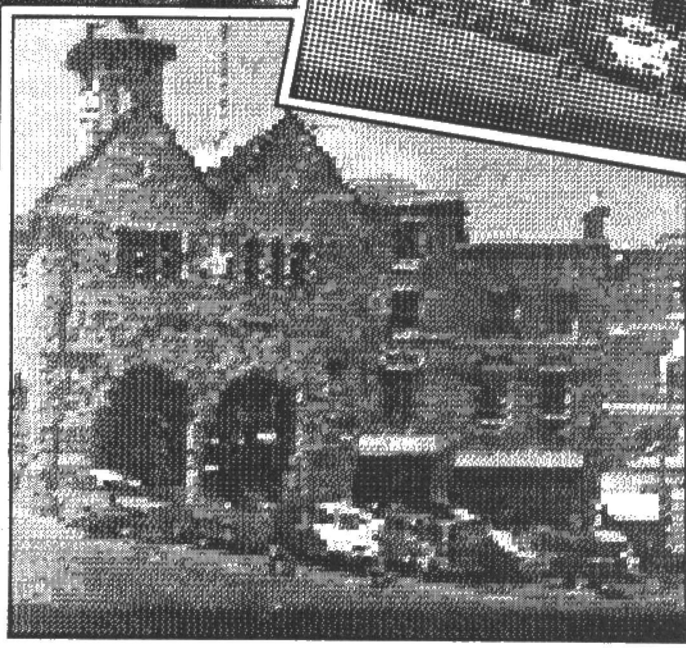


Figure 1. The set of recommended patterns for pixel printing.

Picture 3. The format of this picture is 107 x 128 pixels. The pixel patterns are shown in the Fig. 1.

Picture 4. Ross-On-Wye Market Hall displayed using the pixels described in our Spectrum Image Processor articles (June, July '83).



Picture 5. The same view as picture 4 but this time the display is produced using the pixels of Fig. 1.

E&CM

TABLE 1

GREY LEVEL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
PRINT CHAR.	M	M	0	O	%	O	H	I	★	!	=	;	,	,	.	(space)
OVERPRINT CHAR.	W	O	=	:	(.........						spaces					.........)

TABLE 2

GREY LEVEL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
CHARACTER	@	@	5	5	P	P	7	7	+	+	"	"	'	'	(spaces)	

TABLE 3

STORED DATA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
NEW VALUE	0	1	1	2	3	4	5	7	8	A	B	C	D	E	E	F

If a particular application demands accurate, high speed, analogue to digital conversions, hardware to meet the specification is likely to be expensive. For many applications however, the requirements are less demanding – temperature measurements using a thermistor is a good example of an application where high speed conversion is not required.

In such cases, slower and thus cheaper ADC devices come into their own. The TL507 is such a device. It costs about £1.40 and can be run from both unregulated supplies from 8V to 18V or from a regulated 5V supply.

A/D CONVERTER

Mike Furminger shows how the TL507 can be used as a very low cost ADC system.

If a 5V supply is used, the input range is restricted to values between 1.25V and 3.75V. This can be limiting but, for the low cost of the system, is quite acceptable.

Since the converter works on a single slope comparator the time taken for low voltages to be calculated can be very slow if BASIC is used. The flow diagram shown in **Fig 2** shows the programming of the TL507 operations. These have been converted to 6502 machine code so that suitable programs can be written for the PET, BBC etc. The programs shown will return a count of 128 corresponding to an over range and 0 corresponding to an under range otherwise 127 corresponds to 3.75V and each count will reduce the result by 20mV.

The PET program uses the top end of the second cassette buffer to store the machine code program and the result. The program is called from BASIC with a SYS 950.

The BBC program would appear to be pointless since there are four good ADC channels in the model B. The TL507 however provides a cheap way to run extra channels for control purposes.

Overall this is a one chip system which is cheap enough not to bother about expensive protection circuiting.

E&CM

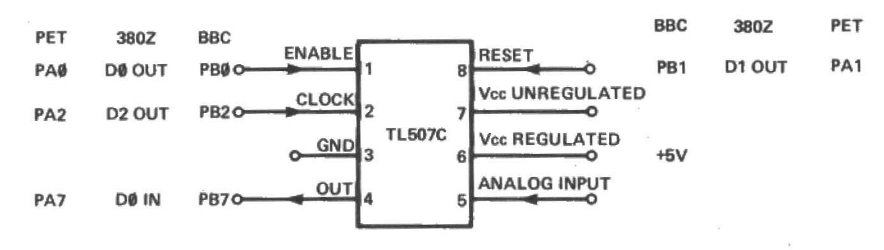


Figure 1. The TL507's pin-out together with I/O connection details for three popular computers.

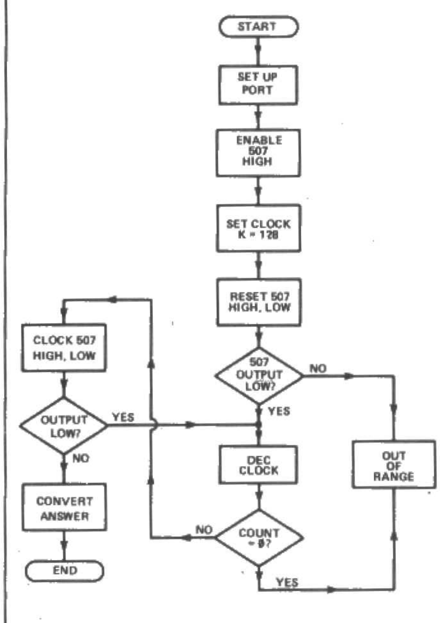


Figure 2. An outline of the operations involved in programming the TL507 for A/D conversions.

BBC program for use with TL507. Although there are four A/Ds in the model B, this program and the IC allow extra channels to be added for control purposes.

```
 10REM TL507 ADC
 20PROCPort
 30PROCassemble
 40PROCadc
 50END
 60DEFPROCPort
 70REM SET UP PORT PB0,PB1,PB2 OUT;PB7 IN
 80PB=&FE60
 90DDRB=&FE62
100ACR=&FE6B
110?ACR=0:REM PB REFLECTS DATA
120ENDPROC
130DEFPROCassemble
140FOR Pass =0 TO 3 STEP3
150P%=&D00
160[OPTPass
170LDA #1:STA PB \enable adc
180LDX #128 \count = 128
190LDA #3:STA PB \ reset
200LDA #1:STA PB
210LDA PB:AND #128 \check for low range
220BEQ out
230.clock DEX
240BEQ out
250 LDA #5:STA PB \clock adc
260 LDA #1:STA PB
270LDA PB:AND #128:BNE clock
280.out TXA
290RTS
300]:NEXT
310ENDPROC
320DEFPROCadc
330R%=USR(&D00) AND 255
340IFR%=0 THEN PRINT"TOO LOW"
350IFR%=128 THEN PRINT"TOO HIGH"
360PRINT"RESULT =";R%
370ENDPROC

>RUN
0D00           OPTPass
0D00 A9 01     LDA #1
0D02 8D 60 FE STA PB \enable adc
0D05 A2 80     LDX #128 \count = 128
0D07 A9 03     LDA #3
0D09 8D 60 FE STA PB \ reset
0D0C A9 01     LDA #1
0D0E 8D 60 FE STA PB
0D11 AD 60 FE LDA PB
0D14 29 80     AND #128 \check for low range
0D16 F0 14     BEQ out
0D18 CA        .clock DEX
0D19 F0 11     BEQ out
0D1B A9 05     LDA #5
0D1D 8D 60 FE STA PB \clock adc
0D20 A9 01     LDA #1
0D22 8D 60 FE STA PB
0D25 AD 60 FE LDA PB
0D28 29 80     AND #128
0D2A D0 EC     BNE clock
0D2C 8A        .out TXA
0D2D 60        RTS
TOO LOW
RESULT =0
```

The TL507's PET software. The program uses the second cassette buffer to store both the program and result.

```
100 REM TL507
110 REM SET UP PORT PA0,1,2 OUT.PA7:IN
120 POKE59459,127
130 REM POKE DATA INTO BUFFER
140 FOR K=0TO48
150 READ D
160 POKE950+K,D
170 NEXT
180 REM  RESULT IS IN 1017
190 REM  RUN ADC WITH SYS950
200 SYS950
210 PRINT"RESULT=";PEEK(1017)
1000 DATA169,1,141,79,232,162,128,169,3
1010 DATA141,79,232,169,1,141,79,232
1020 DATA173,79,232,41,128,240,20,202,240
1030 DATA17,169,5,141,79,232,169,1,141,79
1040 DATA232,173,79,232,41,128,208,236,138
1050 DATA141,249,3,96
READY.
```

ZX Video Techniques Part 1

Michael Graham explains the advantages of direct video connections to computer systems with particular reference to the ZX computers.

The failure of the ZX81 and Spectrum to make available any video output other than the UHF modulated signal has a number of disadvantages. First and foremost, it means that the quality of the display produced by the computers will be limited by the modulation/demodulation processes. The Spectrum suffers particularly in this area. Another drawback imposed by the lack of a direct video output is that neither machine can be easily used in any creative video situations, i.e. combining the computer's output with that of a video recorder either for titling home produced video tapes or for developing interactive programs.

Loss Of Quality

Before going on to describe the ways in which video signals can be extracted from the two Sinclair computers, we'll briefly discuss the reasons behind the loss of quality that results from the process of modulation and subsequent demodulation.

To do this, we'll look at the bandwidth requirements of the Spectrum's 32 column x 22 line display. As a first step it's necessary to appreciate how a screen of information is built up.

The TV screen is scanned in a series of horizontal lines, some 312½ lines are required to build up a full frame (two such Frames go to make up the familiar 625 lines of one field). Starting from the fact that there are 50 scans of the TV tube in one second it's possible to calculate that to scan any one line takes

$$\frac{(1/50)}{312.5} = 64\text{uS}$$

of this line scan only about 50uS is spent forming the display, the rest is taken up by the sync pulse and end of line blanking signals. During this 50uS the 32 characters forming the Spectrum's display must be produced. Each of these characters is formed by an 8 x 8 dot matrix and thus in any one line the number of pixels scanned is 8 x 32 or 256 pixels. From this, it can be calculated that the time taken to form just one pixel is

$$50/256 = 0.195\text{uS}$$

using the relationship that frequency is the inverse of period, this gives a figure for the bandwidth required to adequately resolve the display of

$$\frac{1}{0.195\text{uS}} = 5.13\text{MHz}$$

The calculations above are fairly crude, but the derived figure for the bandwidth requirement is fairly accurate. In relation to the specification of a standard TV set's frequency response, this sort of figure for bandwidth is near the limits of performance of even a high quality set. Certainly many or the cheaper portables do not provide this sort of performance.

The shortcomings in performance are associated almost entirely with the RF sections of the TV set where the bandwidth of the circuitry is only one of many parameters that have to be traded off against each other. In general, there are fewer problems in meeting a 6MHz or greater bandwidth when designing the video driver circuits.

There is an additional difficulty too, in the form of the chroma trap.

Colour Circuits

When colour TV was first introduced to the UK, the brief of the engineers designing the system was to devise a system that would operate within virtually the same bandwidth as a monochrome TV signal. As, in general, it's true that as the amount of information that's required to be transmitted increases, so do the bandwidth requirements increase, it's clear that some tricks were required to transmit colour pictures within the imposed limits.

The solution adopted was to superimpose the colour information onto a subcarrier and, by choosing a suitable subcarrier frequency, cause minimal effects on the monochrome signal. The subcarrier frequency is in fact 4.43MHz and any colour TV set has filters in the signal path to both extract this signal, to reconstitute the colour information, and to remove signals at this frequency from the luminance channel. The latter task is performed by the above mentioned chroma trap.

Removing a band of frequencies around 4.43MHz has an adverse effect on the definition of a display as it is interfering with the high frequency information that the calculations above showed to be essential.

Implications

The inclusion of the chroma trap in all colour TV receivers explains why monochrome TV sets will often give superior results to colour sets. In addition, the bandwidth problems associated with the RF sections of a TV set explain why it is preferable to use a direct video connection to a monitor.

ZX81 Mods

The company Astec have cornered the world market in vision modulators and this is fortunate as it's easy to identify the vision

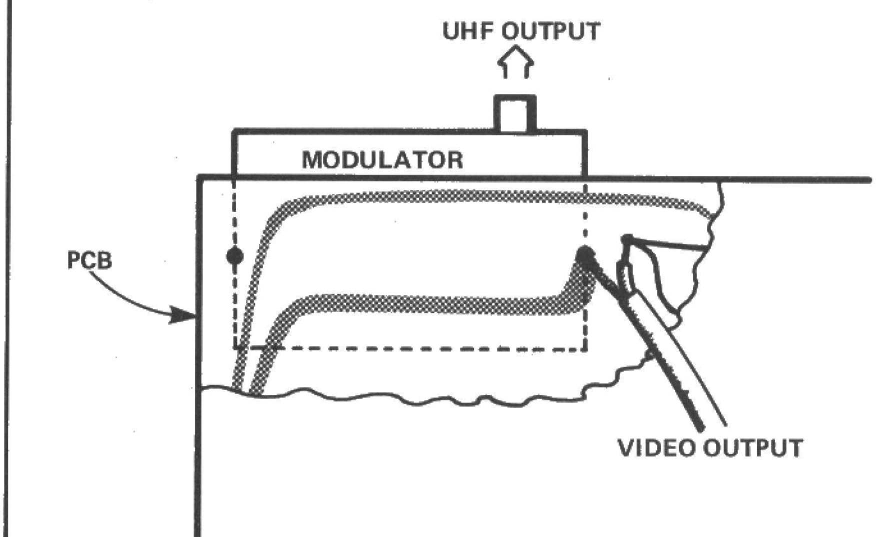
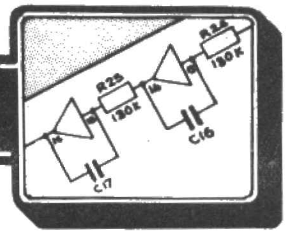


Figure 1. Diagram showing the points on the ZX81's PCB to which the direct video connections are made.



take off points on their standard ch 36 modulator. The signal required by the modulator is also fairly near to a standard composite video signal which again makes the task of extracting a suitable vision signal easier.

Before undertaking this modification it should be borne in mind that it will invalidate any guarantee that may be in existence on the computer.

The first step is to open the Z81 by removing the four screws that hold the case together. To achieve this it will be necessary to remove the rubber feet on the base by prising them off with a sharp knife.

With the bottom half of the case removed, the underside of the ZX81's PCB will be revealed. The tin case of the modulator is easily spotted and **Fig 1** shows the points at which the video signal can be extracted. The best cable to use is a light weight screened audio cable and the screen should be connected to the right hand mounting lug of the modulator. The centre core should be connected to the PCB as shown, the correct point can be verified by making sure that it corresponds to one of the wire connections to the modulator on the PCB's topside.

In most cases, however, the output impedance of the ZX81 is not the standard 75R used in video circuits and with some monitors this may give rise to problems.

The circuit of **Fig 2** can be used to convert the signal to a 75R impedance. This circuit is a straightforward emitter follower that performs the impedance conversion without inversion of the signal.

The circuit can best be built on a small piece of veroboard and can be powered from the ZX81's 9V rail.

Sync Sep

In some applications it may be necessary to separate the sync signals from the video signal. The circuit of **Fig 3** can accomplish this. The circuit is driven by a composite vision signal and, by virtue of the values chosen for the resistors biasing Q1's base, provides an inverted sync pulse at the transistor's collector and a non-inverted video signal at the emitter.

This circuit would be suitable for driving the green input of an RGB monitor requiring separate syncs.

Next Month

This article has shown that it is relatively easy to obtain direct video signals from the ZX81. The problem is lightly more complex when it comes to the Spectrum as some decoding fo the colour signal is required.

Full details next month.

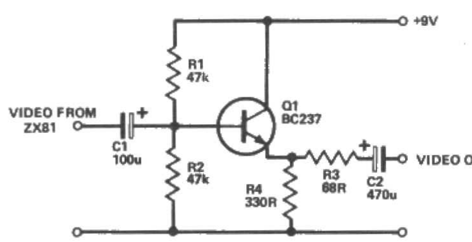


Figure 2. Circuit diagram of the buffer stage that may be necessary in order to convert the high impedance video output of the ZX81 to the standard 75R level.

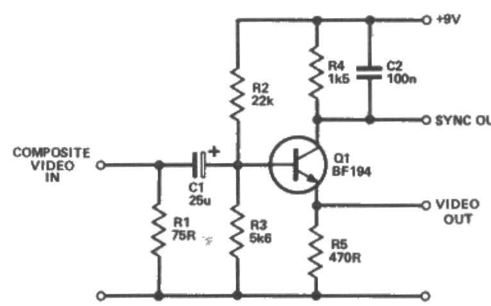
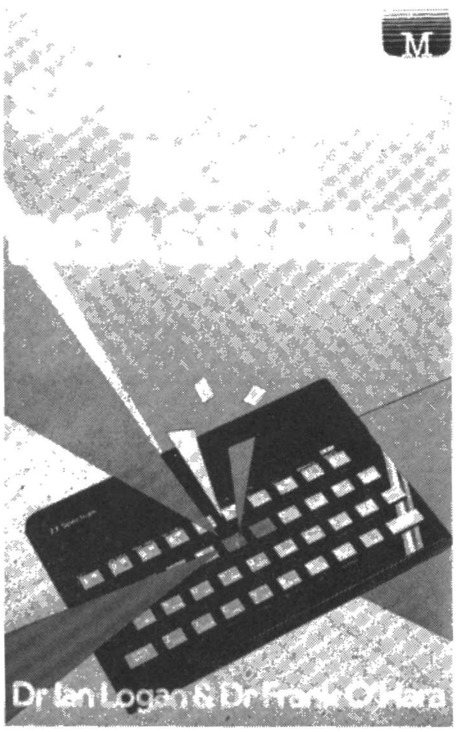


Figure 3. Circuit of the sync separator that may be necessary if the ZX81 is to drive certain types of RGB monitor.

Each month Harry Fairhead will look at a selection of recently published books and software.

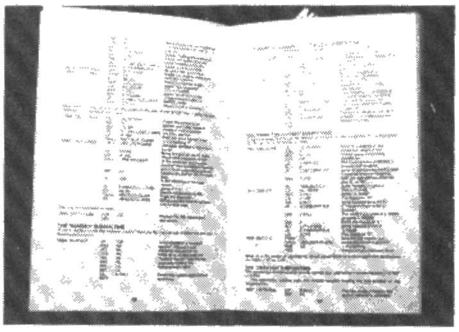
The vast array of books and software currently available presents the microcomputer user with a bewildering choice. This page is intended to help readers by presenting an informed and considered opinion about a couple of books and pieces of software each month. Because space is limited, this review page will be selective, and only books and software that seem to offer a good deal will be included. Every book reviewed on this page can be obtained through the ECM Book Service.

RECOMMENDED READING

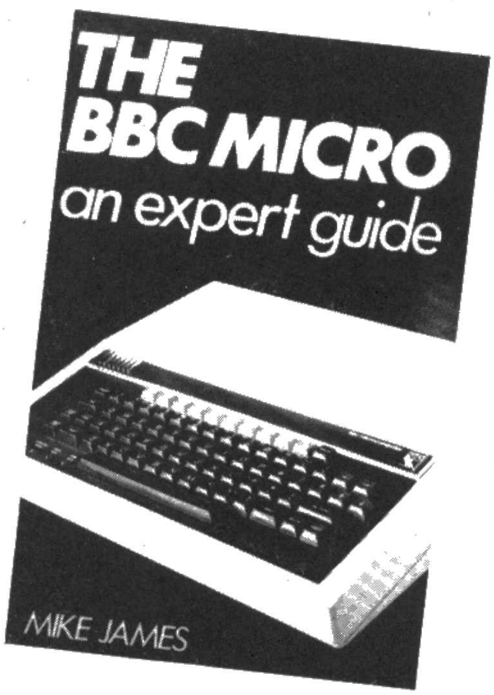


The Complete Spectrum ROM Disassembly
by Dr. Ian Logan & Dr. Frank O'Hara
Melbourne House, 1983.

This book is remarkable for being among one of the longest assembly language listing ever to be published! For this reason alone the Spectrum ROM Disassembly must be of interest to Z80 and other assembly language programmers. The book consists of a commented listing of the 16K ROM program that provides all of the Spectrum's facilities – including ZX BASIC. The listing is presented in fairly logical chunks and the comments do help sort out what is going on where. There is very little general discussion of the overall structure of the program however, and the inclusion of such comments would make it easier and quicker for the newcomer to the ROM to sort out what is going on. In the same way some discussion of the overall hardware structure of the Spectrum would serve to place the software in context. As the ROM disassembly already extends to 236 pages though it is easy to see why these extra items were left out in favour of 'straight listing'! To get anything like the full value from the listing it is important the reader has a good grasp of Z80 assembly language. In addition a rough idea how an interpreter works will help. Even if knowledge of these two areas is not what it should be it will still be possible to examine the machine code subroutines that provide the simpler services to a running BASIC program – e.g. the BEEP subroutine – and possibly find ways of incorporating them into assembly language programs.

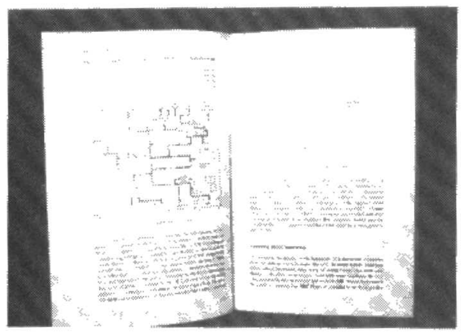


This book is not for the beginner, not even the tyro assembly language programmer. It is a must for any reasonably competent assembly language programmer however, even if the Spectrum is not one of the machines of interest. The reason for this is that the opportunity to examine a BASIC interpreter is rare and not something to be avoided! Even those not contemplating writing or modifying an interpreter will learn much about Z80 programming techniques. The Spectrum owner with an interest in how it works will find anything they need to know about its software is in the ROM disassembly.



The BBC Micro! An Expert Guide
by Mike James
Granada, 1983.

Mike James needs no introduction to readers of *E&CM* and you will probably know his informal yet informed style from his regular articles. Mike is an unashamed fan of the BBC Micro and many will share his enthusiasm. This book is both easy to get into and extremely useful in furthering understanding of the BBC Micro's hardware and software. One of the book's main strength is its presentation of hardware details to explain how to program the BBC Micro. Lots of programming examples are included throughout and many of these are useful as well as illustrative. This book is not for the absolute newcomer to computing. In order to get the best from it the reader should have done at least a little programming of their own and should be familiar with the fundamental ideas of computer hardware. On the other hand, this book is one that will reward reading and re-reading. It contains so



much information that it cannot all be absorbed at once. In common with other computer books from Granada, this book is well produced. It has helpful diagrams and the programs, which are reproduced directly from printer listings, are clear and easy to read. In conclusion, this book combines readability with a clear exposition of highly technical information.

SELECTED SOFTWARE

Logo 2
Cassette
BBC Model B
from Computer Concepts,
16 Wayside,
Chipperfield,
Herts.

LOGO is a simple computer language which has a special place in computer education. It was designed to teach the concepts underlying computer programming, namely, the idea that the computer obeys the commands it is given that the programmer can build up a list of such commands and that commands can be repeated or combined in various ways to produce different effects. LOGO is a language based around the now familiar idea

of turtle graphics. In this case the turtle is a triangle on the TV screen that can be moved around using commands such as FORWARD 100:RIGHT 45 and so on. The tutle can either leave a 'trail' behind him or just move to a new position. In this way patterns and drawings can easily be produced by typing in direct commands and seeing the resulting lines appear immediately. Alternatively, it is possible to define words (essentially subroutines) that can be used to extend the language. There are rudimentary REPEAT and IF statements and these are quite enough to give anyone a taste of programming without getting involved in a full programming language. This implementation of LOGO is easy to begin to use and immediately rewarding. It would indeed be valuable in the classroom. The only reservation is that the manual does not go far enough to enable the novice to use all the facilities of this version of LOGO. It is important to notice that apart from drawing interesting patterns, LOGO is not a language that anyone would use for anything other than education but as such it makes a good introduction to programming for very young children or even adults who have yet to make the acquaintance of a personal computer.

DASM – A Dragon assembler
Cartridge
Dragon 32
from Compusense,
286d Green Lanes,
Palmers Green,
London N13 5TH

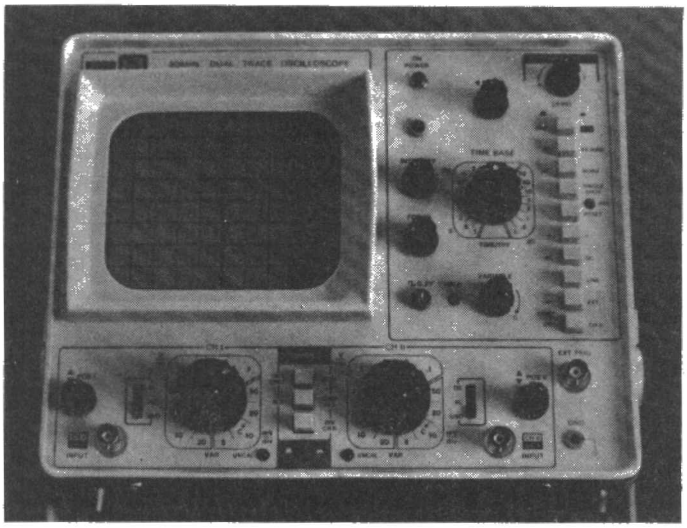
The Dragon is an excellent machine to learn assembly language on because the 6809 that it uses as its CPU is particularly logical and free from quirks of operation. The only trouble is that (unlike the BBC micro) it doesn't come equipped with a built-in assembler. DASM from Compusense is designed to fill this gap, which it does very well indeed. Supplied as a cartridge it is quick to use and eliminates the long waits associated with loading an assembler on tape. The assembly language commands are entered just like a BASIC program. That is, the commands are entered using line numbers, more than one command can be placed on a line using colons as separators and the standard EDIT command can be used to correct errors. In addition the program can be CSAVEd and CLOADed as normal. Indeed, from the Dragon's point of view, the assembly language program looks just like a BASIC program with errors in it! Once assembly language program has been typed, DASM has to be invoked to turn it into machine code. This is done by placing an EXEC &HCFFA just before the assembly language. DASM returns control to BASIC once it has finished and this means that it is possible to place lines of BASIC both before the machine code and after it, thus giving a certain degree of mixed BASIC/assembler programming. DASM is a full two pass assembler with very few restrictions. It uses the Motorola standard mnemonics apart from the change from [] to () and the need to place @ in front of every label.

Although DASM is not a powerful assembler (it has no macro nor conditional assembly facilities) it is easy to use and brings assembly language to the Dragon with almost as much ease as BASIC. The only real complaint is the poor quality of the manual. The manual quite reasonably makes no attempt to teach 6809 assembler as there are a number of good 6809 books on the market. The fact that it is a rather tatty leaflet though lets down a piece of quality software. If a simply easy to use assembler for the Dragon is required however, then DASM will not disappoint you.

E&CM

CROTECH 3337

A full-function dual trace oscilloscope from Crotech-Aplab, reviewed by David Green, B.Sc.



Over the past few years, Crotech Instruments have designed a series of function-specific oscilloscopes, each suiting particular niche in the hobbyist/professional market. For instance, the complete range includes the model 3030 – a 15MHz single beam 'scope with a built-in circuit tester.

The 3337 reviewed here, however, might be thought of more as a general purpose test instrument, since it combines accuracy with a wide variety of features. It is a 30MHz multi-trigger, dual beam 'scope with a basic vertical sensitivity of 5mV per graticule division (i.e. 5mV/cm). The time base is continuously variable – as are the other controls – or calibrated across a range of 0.2us/div to 1s/div, which may be extended using the x5 magnifier push-button. This facility effectively increases the range to between 40ns/div and 2.5s/div – quite useful limits for measuring very high and very low frequency signals.

Composite or alternate triggering is possible and level selection operates in both auto and normal modes. Indeed, the 3337's superficial qualities alone evoked an air of conscientiousness which outweighed its few shortcomings.

Scope For Expansion

The Crotech 3337 has in impressive specification **(Table 1)**, despite its reasonable cost and attention to detail. Accuracy was quoted at 3% for both horizontal and vertical amplifiers, rising to 5% on the sweep ranges (time base). It's a pity, however, that this did not extend to the calibration output, which, whilst being well within the 1% tolerance for amplitude, was outside the quoted frequency of 1kHz, being measured at 670Hz – not very satisfactory for checking your timebase, though to be fair it was not intended for that purpose. The square wave produced was not very clean, with a symmetrical vertical over-shoot, but other parameters checked out perfectly, so possibly this was a fault peculiar to the review sample.

A good feature with this scope is that most of the control functions are accessed via a column of push-buttons down the right hand side. The top three set up operating mode, sweep magnification and auto/normal triggering, with two extra buttons used to activate the 'single shot' and 'reset' for special sweep triggering. This facility was found to be particularly useful, since it was possible to catch an 'event' each time it occurred.

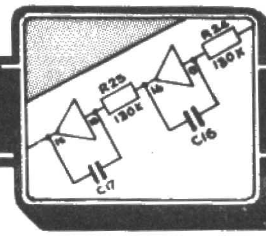
Apart from the usual +/—, AC/DC and INT/EXT switching found on most oscilloscopes, the 3337 has provision for alternate/chop modes on channels I+II *and* I—II (using the invert button), making it an easy matter to sum or difference two signals. Employing this technique, however, highlighted another minor shortcoming – there's no 'beam locate' switch, leaving the operator the task of hunting a trace using the X and Y shift controls. Perhaps, the inclusion of this small luxury would have been of benefit?

Fixed or Variable?

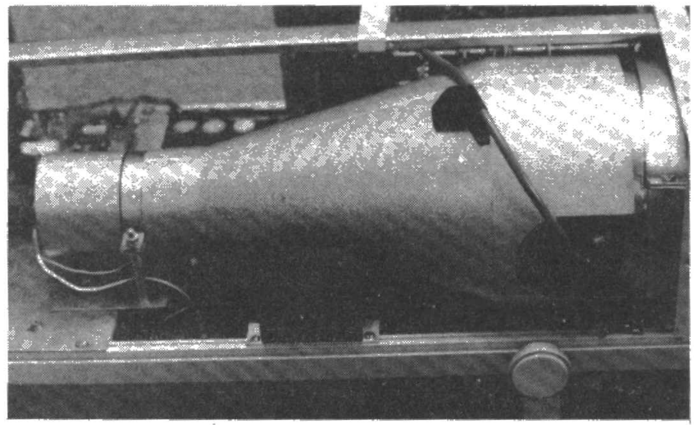
By far the most innovative aspect of the 3337 is the dual control option on all the main potentiometers – the time base and both of the X amplifiers' attenuators. This allows the operator to choose whether he wants accuracy (CAL) or convenience (UNCAL) to take priority. If the latter is the case, for instance when it's necessary to fill the screen with a waveform, then the variable option is selected (a small light shows when the calibrations around the knob no longer apply). Doing this produces a range of settings between the two calibrations nearest on the control knob, and in use, this provision made it very easy to both measure waves, and get the right display for other purposes. Full credit to Crotech for such a design insight.

TABLE 1
Specifications for the Crotech model 3337.

Operation Modes:	Single, dual, add, subtract, X-Y
Sensitivity:	Vertical 5mV/div to 20V/div, continuously variable up to 50V/div. Horizontal 5mV/div to 20V/div in 12 steps (cal). Time base sweep speeds from 40ns/div to 2.5s/div or 0.2us/div toi 1s/div (cal).
Accuracy:	Vertical and Horizontal 3% Time base 5%
Bandwidth:	Vertical DC to 30MHz (—3dB) direct coupled 10Hz to 30MHz (—3dB) AC coupled (rise time = 11.7ns) Horizontal DC to 3MHz (—3dB)
Impedance:	1M in parallel with 35pF (Vert and Horiz)
Cal Out:	200mV peak-to-peak at 1kHz (approx)



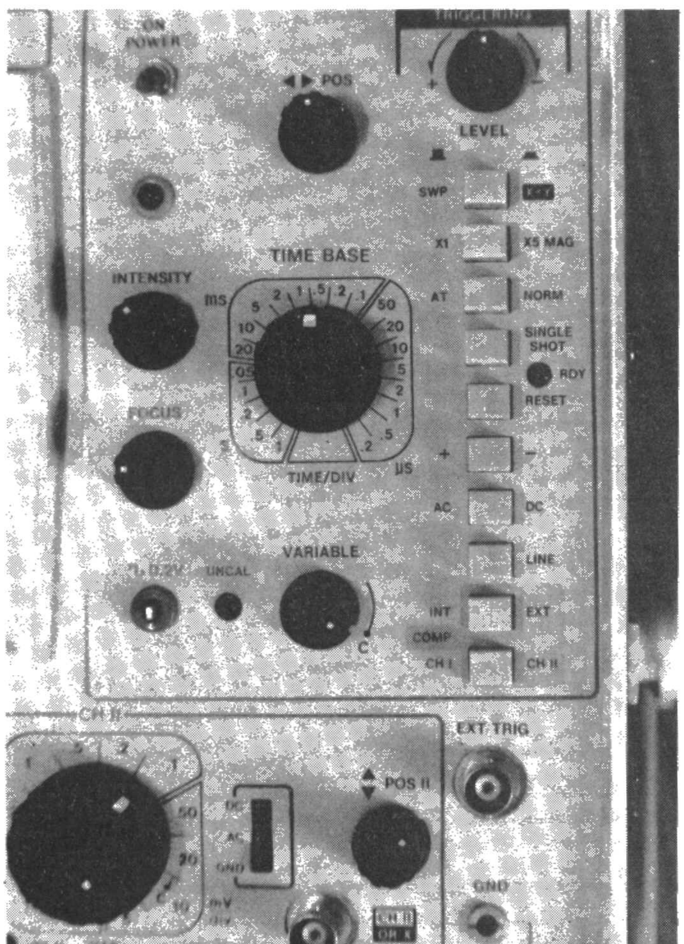
In conjuction with the above dual measurement modes, the basic sensitivity of both X amplifiers and the time base scale is certainly wide enough to cover a host of applications. The time base, however, has the additional enhancement of a x5 multiplier on the vertical push-button column. So, should you find it desirable to lengthen a wave on the screen, the button does it without the need to change the timebase settings. Just one more example of good design sense one of the qualities of the 3337 is – once a wave has been positioned on the screen, it is possible to make all the required measurements and analyse them with a minimum of changes in the settings of the controls.



What's On Display

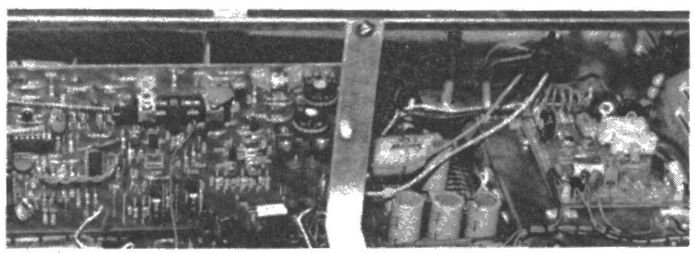
As can be seen from the photograph, the cathode ray tube used in the scope, despite emerging onto a rectangular graticule, is terminated by a circular face (13cm diameter). It has a yellowish-green phosphor (P31), which, with the 3337 requires an accelerating voltage of about 10,000 volts. Both traces were extremely clear as a result, though there was a tendency for 'bowing' towards the centre.

Signals being displayed on the screens could be shifted off the graticule in both vertical and horizontal directions, but it was not possible to change the width of either trace without adjusting an internal preset – a small irritation when you wish to study the extreme ends of a wave.



The real challenge for a scope which boasts a frequency response up to 30MHz, lies in its ability to display traces at the top of the range without loss of clarity. In this respect the 3337 performed as well as scopes costing at least twice as much. Signals retained a clean appearance and were well defined even with the x5 multiplier switched in.

At the other end of the scale, low frequency signals were just as easy to display. The traces were neither 'blotchy' nor distorted in any fashion and the flyback was undetectable – quite a feat at low sweep speeds. All of the amplitude/shift controls functioned perfectly right across the time base range and waves were easy to stabilize on the screen regardless of sensitivity settings. As mentioned earlier, there's no doubt that Crotech have spent a lot of time making sure that the 3337 can be set, then left, without recourse to constant re-adjustment due to drift.

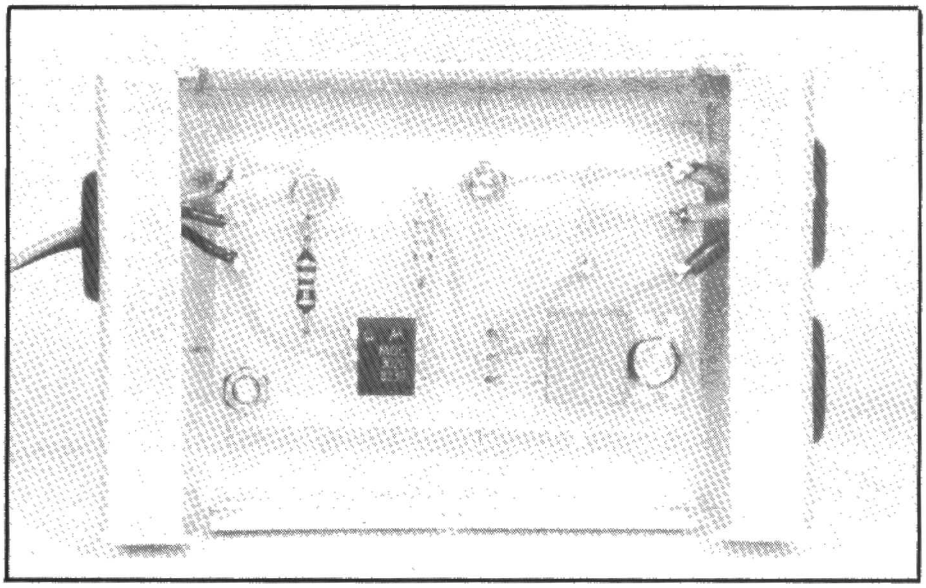


A Few Last Words

After spending several hours working with the 3337, it has become somewhat of a permanent addition to my test bench (pity Crotech wanted it back!). Even taking into account the few minor points which, with a bit of attention, could be resolved, the scope was found to excel at providing exactly the right sort of information with the minimum of adjustment. Different modes are organised so that altering a particular parameter is merely a case of switching in or out a single button. So, if you are thinking of buying a dual beam scope with all the facilities and none of the fuss, then the 3337 should definitely be on your list. It comes complete with a pair of modest, yet accurate, test leads and costs £●●●.

The 3337 is manufactured by Crotech-Aplab and sold by the Electronics & Computer Workshop, Chelmsford, Essex.

E&CM



BBC DARKROOM TIMER

The BBC model B with its in-built timer and latched output port is ideally suited to a range of control applications. John Baker has designed a Darkroom Timer that makes full use of the computer's facilities. He also describes mains isolation techniques in general.

The main requirement of any interface is that there should be no direct connection of any kind between the computer and any mains powered equipment so that the computer remains isolated from the mains supply. The simplest form of interface is a relay but unfortunately, relays that are capable of operating from the 5 volt output from a microcomputer either require a low operating current but are incapable of handling 240 volts A.C., or are capable of handling powerful mains loads but require high operating currents.

A relay is a perfectly satisfactory method of interfacing where a suitable power source is available, and the circuit diagram of **Fig 1** shows a circuit that works well with the BBC model B microcomputer. Q1 is used as a common emitter switching transistor with R1 giving current limiting (where necessary) in the base circuit. The relay coil is driven by the collector of Q1, and D1 is the normal protection diode which suppresses the high voltage spike which would otherwise be generated across the relay coil each time it was switched off. A set of changeover relay contacts is used here as a set of normally open contacts which connect the "live" mains supply through to the output when the PBO input is at logic 1 and the relay is activated.

The circuit requires a 12 volt supply with a supply current of about 40 milliamps when the relay is activated. The BBC model B microcomputer has a suitable supply output, but with most other machines it would be necessary to add a 12 volt power supply using a step-down and isolation transformer. The relay is an Omron type having a 12 volt 306 ohm coil and contacts rated at 8 amps with a 240 volt AC supply – this is available from Maplin Electronic Supplies Ltd.

Opto-Isolator

There are alternatives to using a relay these days, and many would no doubt regard a relay as being old fashioned and out of date. In fact a relay often represents the best solution to switching problems due to its inherent isolation between the controlling and controlled equipment, and because no significant power is lost through the switching element of the device. Semiconductor switching devices tend to dissipate a few watts when handling fairly large currents, and while this loss of power is unlikely to have any noticeable effect on the controlled equipment it does give the problem of heat-sinking the switching device to prevent over-heating.

A triac is the semiconductor device normally used for controlling AC loads, and this is used in the manner shown in **Fig 2.** A bias current of typically about 20 milliamps (and either polarity) is applied to the gate and MT1 terminals, and the device then conducts between its MT1 and MT2 terminals with a voltage drop of only about 1.2 volts through the device. Once biased into conduction the device remains in that state even if the input bias is removed, but only if the current flow between the MT1 and MT2 terminals remains above a certain level (usually about 30 milliamps). When an AC signal is being controlled the MT1 to MT2 current falls to zero at the end of each half cycle, and for most practical purposes the triac can be regarded as switched on when the bias current is present and switched off when the bias is absent.

It would not be safe to operate a triac direct from a micro computer since one side of the mains supply would be connected to the output of the microcomputer. Some form of isolation circuit must therefore be used between the micro and the triac, and an opto-isolator is probably the most convenient way

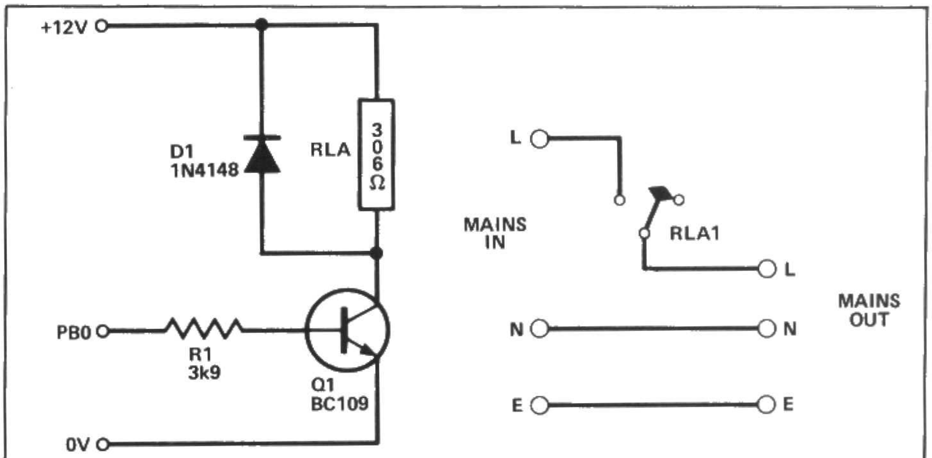


Figure 1. A relay circuit that can be used to control mains powered equipment with the BBC model B computer.

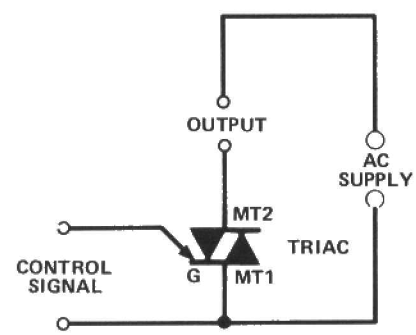


Figure 2. A basic Triac circuit. Applying a low voltage current of about 20 mA to the gate terminal of the device will allow mains current to flow between the MT1 and MT2 terminals.

of doing this. These days there are various types of opto-isolator, but they all consist of a light emitting diode (LED) with its output directed at a photosensitive semiconductor device of some kind. In the early devices the photosensitive device was a photodiode of phototransistor, but in more recent devices photothyristors and phototriacs are used. Opto-isolators are contained in an opaque encapsulation so that ambient light is kept away from the photosensitive element.

The basic idea is that with no input to the LED the photosensitive element is in complete darkness, and whatever type of semiconductor device is used, it is switched off. If the LED is switched on, the light received by the photosensitive element results in a strong leakage current being passed, and this is effectively the same as biasing the device into conduction in the normal way. There is no direct connection from the LED to the photosensitive element however and the required isolation is thus obtained. Most devices can withstand at least 1500 volts without breaking down and passing a current from the LED side of the component to the detector side, which is more than adequate for use with the 240 volt AC mains supply which has a normal peak level of just 340 volts.

Practical Circuit

In some cases it might be possible to simply feed the LED of a triac opto-isolator from the output port of microcomputer with the triac side being connected in series with the "live" mains lead to the controlled equipment. This will not usually be possible though as the output current from the microcomputer will often be inadequate to drive the LED of the opto-isolator properly. An opto-isolator actually only needs a drive current of a few milliamps, but the output current from the user port of a micro can be very limited indeed. A test on the BBC model B micro, for example, gave a LED current of just over 1.5 milliamps. Another problem is that the only triac opto-isolator which is readily available at present can control a maximum (RMS) current of 100 milliamps, which represents a maximum power of just 24 watts with a 240 volt mains supply. This would be inadequate for most applications.

These problems can be overcome using the circuit of **Fig. 3.** Here an emitter follower buffer stage (Q1) plus current limiting resistor (R1) are used to provide an adequate LED drive current, and the triac section of

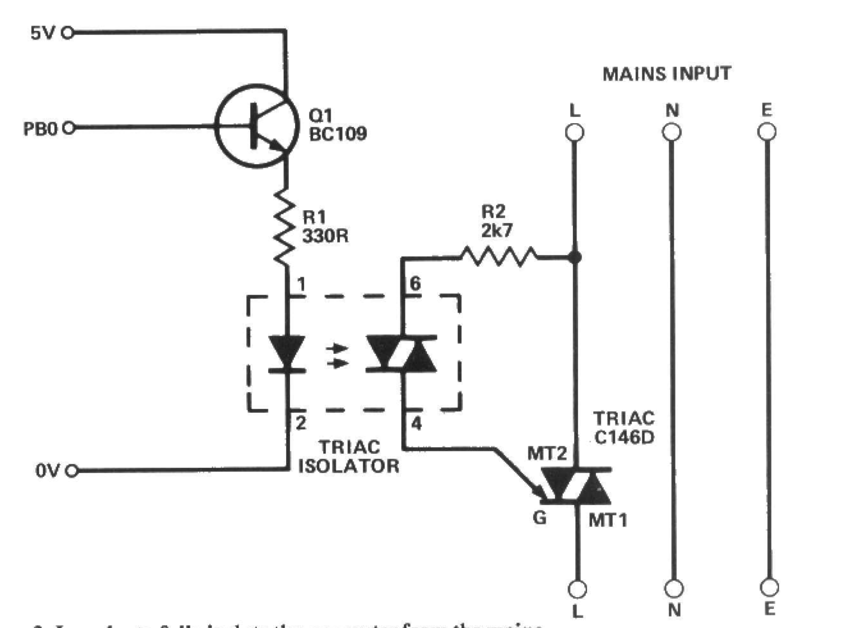


Figure 3. In order to fully isolate the computer from the mains supply it is necessary to use a Triac Opto-Isolator. Transistor Q1 buffers the output port and, together with R1, ensures adequate drive current.

the opto-isolator is used to switch on a higher power triac when the opto-isolator is activated. The C146D triac has a current rating of 10 amps which gives a maximum load power of 2400 watts using the 240 volt mains supply. However, the triac would need to have a substantial heatsink in order to control powers of this order, and about 500 watts would be a more practical power limit, or around 750 watts if the triac is fitted with a small bolt-on heatsink.

The circuit can be driven from any normal MOS or CMOS output which is operated from a supply of about 5 to 10 volts. TTL outputs have a much lower logic 1 output voltage, and it might be necessary to alter the value of R1 and make other modifications to the input circuit in order to obtain satisfactory results with TTL circuits.

The Software

For an enlarging exposure timer to be of any practical use it is important that it allows an exposure time to be repeated easily: that it allows the lamp to be switched on and off at will for docusing and composing purposes, and that it allows the exposure time to be set in small increments. It is also very desirable for the timer to indicate how much of the exposure time has elapsed, as this aids the techniques known colloquially as 'burning in' and 'dodging'.

This program does all that, and in addition it allows the exposure to be interrupted part-way through, and re-started, as often as required, whilst maintaining the correct overall time. If that were not enough, it also calculates the exposure changes required when changing magnification, or when changing filters in colour printing!

This last feature is obtained by accepting a string rather than a number at line 50. This is then converted into a number by the EVAL function in line 55. The easiest way to calculate the change of exposure time for a change in magnification is by the formula:

Construction

A suitable PCB design for the interface is shown in **Fig 4,** and this board is produced using the normal techniques. The triac is bolted in place using a 6mm M3 screw plus fixing nut, and it is then soldered into circuit. Make quite sure that the triac opto-isolator is fitted onto the board the right way round. It will not be damaged if it is mounted the wrong way round, but it might be difficult to remove it and reconnect it properly. Pin 1 is indicated by a small dot or indentation on the body of the component, as for other DIL encapsulated devices.

The board can be fitted into a small aluminium box measuring about 76 by 51 by 25mm. It is mounted using 6BA or M3 bolts about 12mm long with 6mm spacers between the board and the case to prevent the connections on the underside of the board from short circuiting through the case. The case must be earthed via the mains input lead for safety reasons, but this connection is provided by way of one mounting bolt and spacer. One end of the case is drilled to take the three way input cable (which is terminated in a plug of the appropriate type for micro concerned) and the other end is drilled to take the mains input and output cables. All three holes should be fitted with grommets to protect the cables.

Of course, if preferred the unit could be fitted into a larger case so that a mains outlet could be mounted on the case, and any piece of equipment of acceptable power rating could then be controlled via the interface. If several items of equipment are to be controlled by the micro then it will be necessary to build up an interface board for each piece of controlled equipment, but these can all be mounted in the same case if desired.

Together with a suitable micro an interface of this type must have almost limitless applications. The prototype was designed for use with the BBC model B micro as an advanced enlarger timer, and a suitable program (plus programme notes) for this application follow.

Obviously the mechanical construction of the unit must be varied to suit individual requirements.

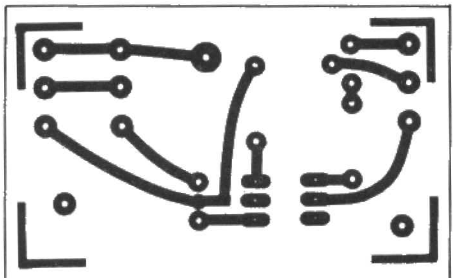


Figure 4. The full size foil pattern of the opto-isolators PCB.

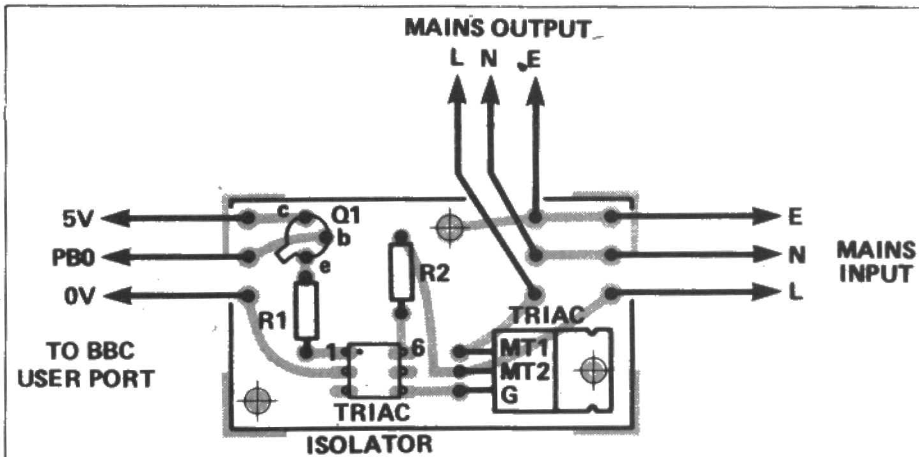


Figure 5. The overlay of the opto-isolator project. When building the project take care to ensure that all mains wiring is well insulated.

---

new time = old time/old height 2*
new height 2

The old exposure time is stored in the variable "runtime" in this program, and as it is likely to be needed often, function Key 0 is programmed to print this by line 20. Unfortunately it is the variable name, not the value, that appears on the screen. The above calculation would thus appear as:

Exposure Time?runtime/20 2*16 2

(these numbers of course are just for demonstration) when RETURN is pressed the new exposure time will appear.

Changes in filtration are calculated by multiplying the old exposure time by the factors of any filtration taken out, and dividing by the factors of any filters taken out. Both calculations can be done at the same time, provided the maximum acceptable string length is not exceeded.

It is also possible, of course, to simply input a completely new exposure time.

Line 40 sets the Data Direction Register B to output. Once this is done, setting ?&FE60=1 will turn the lamp on and setting ?&FE60=0 will turn it off. The output can be made to change state by setting ?&FE60=?&FE60+1, and this is used at lines 110 and 340 to give 'focus switch' control.

Lines 180-220 are a loop, continuously comparing the elapsed time since the exposure was started with the required exposure time. When the exposure is completed, the program branches to line 290, which invites one to repeat or reset the exposure, or to turn the lamp on or off for focusing. (Lines 300-360) Line 370 takes you back to line 290 if any key other than 'R', 'S', or 'F' is pressed.

It is possible to interrupt the loop at any time by pressing the space bar. In this case, line 240 turns the lamp off, and line 230 puts the TIME reading at the time the bar was pressed into the variable "hold". When the exposure is restarted by again pressing the space bar, line 270 resets TIME to this value, and line 280 sends the program back into the loop.

As darkroom work is done largely in the dark, comprehensive error handling was considered necessary. In general, only pressing the space bar or BREAK will

PARTS LIST.

Resistors (all 1/3W, 5%)

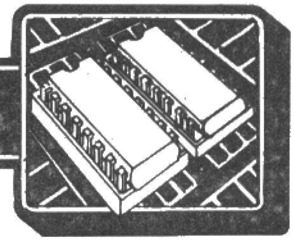
R1	330R
R2	2k7

Semiconductors

Tr1	BC109
Triac	C146D (400 PIV 10A)
Triac opto-isolator	(Maplin Electronic Supplies Ltd.)

Miscellaneous

PCB, aluminium box, cables, grommets, fixings, etc.



interrupt an exposure. Line 410 effectively disables the ESCAPE key. Should this line send the program back into the loop, a "no FOR" error will be generated (ERR=43). In this case, line 400 will cause a return to the start of the loop.

If ESCAPE is pressed during the early part of the program, whilst waiting to start the exposure after a set or reset, line 390 will send the program to line 290, which as previously mentioned invites one to repeat or reset. This is a good escape route if an excessively long exposure is accidentally or inadvertantly set. If the ESCAPE key is accidentally pressed at this stage, simply pressing 'S' will send you back to where you were.

If the exposure time entered, or calculated, is too large for the computer to handle, this is dealt with by line 380. Unfortunately, the old exposure time is lost.

Photographers will realise that, in order to use a television or monitor in the darkroom it will be necessary to filter the light from the screen to prevent 'fogging'. Alternatively, the screen can be covered or the VDU turned off while sensitive materials are being handled.

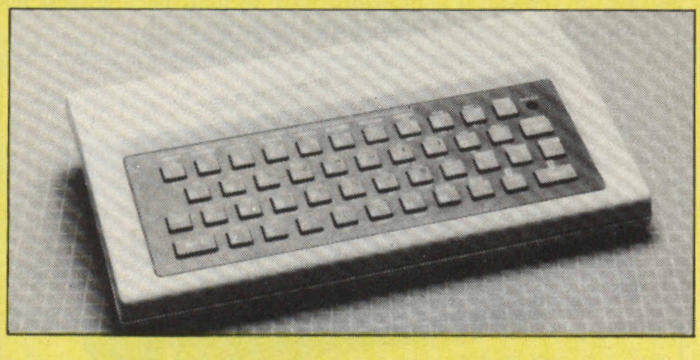
E&CM

```
 10  ON ERROR GOTO 380
 20  *KEY 0 runtime
 30  CLS
 40  ?&FE62=&01
 50  INPUT TAB(2,2);"Exposure time ",runtime$
 55  runtime=EVAL(runtime$)
 60  IF runtime<0.1 OR runtime>251 THEN CLS:GOTO 50
 70  CLS:PRINT TAB(2,2)"Exposure Time ";runtime;" Seconds"
 80  PRINT TAB(2,4);"Press Space Bar to Start"
 90  PRINT TAB(2,5);"or 'F' to focus/off"
100  select=GET
110  IF select=&46 THEN ?&FE60=?&FE60+1:GOTO 100
120  IF select=32 THEN GOTO 130 ELSE GOTO 100
130  PRINT TAB(10,7);"Press Space Bar"
140  PRINT TAB(10,8);"to Hold/Restart"
150  ?&FE60=1
160  T=TIME:VDU 23,1,0;0;0;0;
170  PRINT TAB(10,10);"Elapsed Time"
180  REPEAT
190    elapstime=(TIME-T)/100
200   .PRINT TAB(10,12);elapstime
210    IF elapstime>=runtime THEN ?&FE60=0:GOTO 290
220    UNTIL INKEY(0)=32
230  hold=TIME
240  ?&FE60=0
250  REPEAT UNTIL GET=32
260  ?&FE60=1
270  TIME=hold
280  GOTO 180
290  CLS:VDU 23,1,1;0;0;0;
300  PRINT TAB(10,6);"Press'R'to Repeat"
310  PRINT TAB(10,8);"or 'S'to Reset Time"
320  PRINT TAB(10,10);"or 'F' to Focus/off"
330  repeat=GET
340  IF repeat=&46 THEN ?&FE60=?&FE60+1
350  IF repeat=82 THEN GOTO 70
360  IF repeat=83 THEN GOTO 30
370  GOTO 290
380  IF ERR=20 THEN CLS:GOTO 50
385  IF ERL<=100 THEN GOTO 290
390  IF ERR=17 GOTO ERL
400  IF ERR=43 THEN GOTO 180.
410  CLS:REPORT:PRINT" at line ";ERL
```

The full program listing of the Darkroom Timer's software.

LASER 200 & COMX 35

Ken Alexander looks at two low cost machines from the Far East. Will they offer any real challenge to the Spectrum? Read his review to find out.



The Comx 35 and the Laser 200 are both low cost colour computers and are both distributed by Computers For All, but the machines themselves are two very different items of hardware. The Comx 35 is a £120 computer with a generous 35K of RAM and the unique feature of a Joystick incorporated into the keyboard. The Laser breaks the £100 barrier by a long way and offers colour computing at a price of £70.

As it's the Laser that is likely to demand most attention, we'll begin this double barrelled review with a look at the machine that slits nicely between the ZX81 in terms of price and, as we will see, in some ways performance.

Value For Money

As one of the most attractive features of the Z80 based Laser is bound to be its price, it's probably best to begin this review with a brief look at exactly what the £70 purchase price brings in the way of features.

Included in the basic specification are colour graphics, a decent sound generating facility, a 600 baud cassette interface and floating point BASIC interpreter – what you don't get is a generous amount of memory. The standard machine has 4K of RAM on board and augmenting this with another 16K of RAM adds £29.95 to the price. Thus a Laser system with a reasonable amount of memory (20K) costs all but a pound more than the Spectrum and this point should be firmly bourne in mind when assessing the merits of the Laser.

First Impressions

The Laser is an attractive machine with a keyboard that is closer to that of the Sord M5 than it is to the Spectrum being a moving key design, the keys themselves being formed from a hard rubber material.

The keyboard had a positive feel to it and each keypress is acknowledged by a audible beep. The Laser 200 also provides an auto repeat function.

A single keyword entry system is supported, pressing the CTRL key together with one other key will enter the majority of BASIC commands. The designers of the Laser have chosen to associate the most used keywords with the keys along the top row of the machine rather than with the alpha key corresponding to a command's initial letter. Thus CSAVE, CLOAD, LIST, RUN and VERIFY plus a couple of others, are to be found above the line of numeric keys. This seemed a very sensible idea and meant that the initial process of finding the way around the keyboard was made far easier.

The single key facility, while speeding up entry, does not save any memory, as the commands are stored as ASCII strings and not as a token when entered in this fashion.

The I/O facilities provided on the Laser are fairly comprehensive, although, having said that, it must be pointed out that there is a complete lack of any detailed description of the expansion ports in the literature supplied with the machine.

The full list of I/O facilities is though, Memory Expansion – this, it is assumed, makes available all the data, address and control signals of the Laser, Peripheral – this is described as the place to connect a Printer Interface module or a Joystick interface. This port has less connections than the Memory Expansion Port and its possible that some internal decoding makes this a very convenient way of getting data to and from the Laser but without much more information than that shown in the manual (none) it is impossible to use.

Other connectors along the rear panel are the cassette I/O (no remote control option), the power input, the VHF video output and a welcome video output for use with a monitor. This is a composite video output and not an RGB type which may have been preferable.

Along the right hand side of the machine is an on/off switch – something that should be fitted as a matter of course to all computers.

Using the Laser

Switching the Laser on brings a fairly terse READY message to the screen with a flashing green cursor.

Syntax is checked on a line by line basis as a program is entered and editing is accomplished via straightforward although old fashioned line editor with the cursor being positioned by the four cursor control keys and characters deleted and inserted by using the RUBOUT key or by putting the Laser into the insert mode by entering CTRL + L.

A full list of the commands provided by the Laser's BASIC is shown in **Table 1** and most of these require little comment. The graphics and sound capabilities of a computer of the Laser's price will, however, be of major concern to would-be users and we will look at these in a little more detail.

Sounds Good

It may seem common sense to position any loudspeaker designed to output noise from a computer on any surface of the machine apart from its underside where the sound will almost certainly be muffled by whatever surface the computer is being used on. Common sense maybe, but it is surprising just how many computers have their speaker in this position. Not so the Laser, we're pleased to report.

The Laser's sound command is fairly primitive but, having said that, does what it is supposed to do.

A total of 31 notes can be sounded at one of nine different note durations. The command takes the form of

SOUND P,D

where P is a number from 1 to 31 determining the note's pitch and D is again a number, this time between 1 and 9, that selects the notes duration.

Used in conjuction with the DATA statement that is supported by the Laser, simple tunes can be readily produced by the machine.

Graphics

The graphics ability, or rather lack of it, is perhaps the Laser's weakest point. On power-up the Laser is in text mode, the display being 32 characters x 16 lines. In this mode, ZX81 "chunky-style" graphics, are available in any one of the nine available colours on a black background. Text however can only be displayed as orange on a green background or as yellow on green.

The "Hi-Res" mode allows greater, though by no means spectacular, definition displays with 128 x 64 resolution. In this mode the colours available are restricted to two sets chosen from the complete palate. These are a buff border with orange, magenta, cyan or buff graphics and a green boarder with a red, blue, yellow or green display.

TABLE 1

Arithmetic functions:	**Program statements**
SQR	DIM — Dimensions
INT	STOP
RND	END
ABS	GOTO
SGN	GOSUB
COS	RETURN
SIN	FOR . . . TO . . . STEP
EXP	NEXT
TAN	REM
LOG	IF . . . THEN . . . ELSE
ATN	INPUT
	PRINT
String functions:	PRINT TAB
LEN —	PRINT USING
STR$	PRINT @
VAL —	LET
ASC —	DATA
CHR$	READ
LEFT$	RESTORE
MID$ —	
RIGHT$	**Commands:**
INKEY$	LIST
	RUN
Logical operators	NEW
AND	CONT
OR	VERIFY
NOT	CLOAD
	CSAVE
Graphics and sound	CRUN
functions:	CTRL RESET
CLS —	
SET —	**Other statements**
RESET	PEEK
POINT	POKE
COLOR	LPRINT
SOUND	LLIST
MODE	INP
	OUT
	COPY
	USR

Together with the usual plot commands the displays that can be produced are at best crude and indeed, the pre-recorded games software supplied with the machine, in the main restricted themselves to textual displays and the only graphics based game supplied was a very 'chunky' version of the arcade breakout game.

Saving and Loading

The cassette loading and saving system runs at 600 baud, not the fastest but by no means the slowest of cassette interfaces and it worked reliably once the correct replay volume level had been ascertained.

A useful VERIFY facility is provided as well as the ability to auto run a loaded program with the CRUN command.

Expansion

The 16K RAM pack mentioned above slots into the expansion interface and is a well thought out design that, when in position is ell supported by four feet and is thus unlikely to suffer from the dreaded wobble of the ZX81 expansions.

In order to use a printer with the Laser it is necessary to use a printer interface but this was not available for review.

Conclusion

As we pointed out at the start of this review, the Laser plus RAM pack costs about the same as a 16K Spectrum. In some areas, notably the sound facility, the Laser can out-perform the Spectrum but Sir Clive's machine has the edge in most respects.

With the wealth of software, hardware and publications aimed at the Spectrum user the Laser has some catching up to do. Whether or not the Laser has a role as a ZX81 type machine for the person who wants to begin computing for the least possible cost but wants something more than the very basic performance of the 81 remains to be seen. In the end it is probably down to how much interest the various software houses show in the machine.

Comx 35

As we suggested above, the Comx is a very different machine to the Laser, in fact its different to nearly every computer on the market. The difference lies in the fact that the machine is based on the RCA 1802 processor. This MPU has rarely seen the light of day in this country (about the only system based around it was the ELF) and while this will not worry those who wish to stick with the computer's BASIC, anybody wishing to program in machine code will have to learn a whole new language. This will also mean that software houses will be slow to produce software for the machine, if indeed they get round to it at all.

This introduction is not meant to be too negative however as the Comx has a number of attractive features, perhaps the most evident being the inclusion of a joystick as part of the standard machine. More on the joystick later.

Initial Impressions

At first sight, the impression given by the Comx is PLASTIC. The case is plastic and even the 55 keys are fabricated from a hard plastic. To this reviewer's mind at least, this gives the Comx a cheaper appearance than the Laser that sells for £50 less than the Comx. Yet another negative, which to be fair, should be countered by stating that the keyboard has a nice feel to it. The concave keys mean that in use, the typing finger is gently guided to the centre of the key. This means, that although the keyboard is rather small, it is possible to enter data accurately and at speed. The keyboard is also blessed with an auto repeat function.

The back of the Comx's case features the input and output sockets for the loading and saving of programs on a recorder (two 3.5mm jack sockets), a Phono socket outlet for the UHF modulated video signal, the power input socket and, as on the Laser, a welcome on/off switch.

The right hand side of the case is taken up by the computer's expansion connector. As with the Laser, that's about all we can say about the connector, as the manual gives absolutely no information as to the function of the various lines. Together with the absence of system memory maps, machine code entry points and the like, life for the machine code programmer or those wanting to hook the computer up to other items of equipment is made difficult, if not impossible.

The Comx comes complete with a fairly generous complement of memory with 32K of user RAM and 3K of video memory ma' 'ng of the 35 of the machine's title. To put this in context though it is necessary to refer once again to the omnipresent Spectrum that for an extra £5, makes 48K of RAM available for the video display/user programs.

Up And Running

Switching on the Comx produces a series of musical notes, proving the computer can make sounds, and a couple of colourful alternating turn-on messages. These can be used to adjust the TV set for the best possible display and, if acceptable result cannot be obtained, two presets accessible via two holes in the base of the machine, allow fine tuning of the Sync and video levels of the video output.

The review Comx produced a rock solid and well defined display of a 40 x 24 line format. The character set available on power up is the standard ASCII upper case set plus numerics and the miscellaneous punctuation symbols and $, # symbols plus a set of PET/Sharp like graphic symbols. Each of the symbols is, however, redefinable by way of the SHAPE command that takes the form

SHAPE (X,"18 hex numbers")

Here X is an integer that corresponds to the decimal code of the key and the 18 hex numbers determine which of the 54 pixels that go to make up each character cell are ON and which are OFF. Each row of the redefined shape can be defined as one of four different colours.

Colour Capabilities

In general, the colour capabilities of the Comx are rather unusual. On power up, all text entered from the keyboard is displayed in white while any messages generated by the machine are produced in cyan. The manual says this is to produce a feeling of dialogue with the machine, and this multi-coloured display is indeed helpful when analysing a screen full of text.

These two colours can be altered by the command

COLOUR (X)

where X is a number between 1 and 12 and selects from a set of colour pairs that again differentiate between keyboard input and computer output text.

The background colour of the screen can also be altered with the. command

SCREEN (X)

with X being a numeric in the range 1-8.

The last command associated with the Comx colour display is CTONE. This facility is either ON or OFF and when enabled causes the brightness of characters displayed on the screen to be increased.

From the above, its apparent that the Comx possesses a versatile graphics and colour capability that, with patience on behalf of the programmer, the computer should be capable of producing some excellent graphic displays.

Sound System

The sound section of the Comx is similarly fairly comprehensive with the main command taking the form of

MUSIC (X,Y,Z)

Here X determines the note to be sounded in any particular octave, Y determines the octave (one of the eight available) and Z selects the amplitude on a scale from 1 to 15.

Invoking the MUSIC command will enable the tone until it is turned off (or modified) with a second statement. As with the Oric, a MUSIC (0,0,0) command should be readily accessible.

Other commands associated with sound generation are

TONE (X,Y,Z)

This is very similar to the music command with the variables Y and Z taking on the same function but here X offers a finer control of the frequency generated within the selected octave and can take any value between 1 and 128.

The NOISE command generates noise in a selected band frequencies. It takes the form of

NOISE (Y,Z)

with Y selecting one of the eight frequency bands and Z varying the amplitude.

Finally, the command

VOLUME (X)

will give overall control of the volume from the Comx. X can take one of four values.

Although the Comx does not feature the three tone channels featured on machines such as the Oric, it is quite versatile, and again with patience at the programming stage should yield most of the Zaps and Pings demanded of it.

It's a Joy

The joystick is a fairly simple, but effective control. It generates the equivalent of key presses, with the numbers from 136-139 being generated for its four decodeable directions. It can be read with the BASIC KEY command and its incorporation is to be applauded – other manufacturers please note. It's disappointing though, that the Comx does not provide a port for a second joystick to be added to the machine.

BASIC Facilities

The BASIC of the Comx has some good and some bad points but considering the world shortage of 180Z BASICs, Comx have come up with a good workman-like interpreter.

All the expected commands are there together with such goodies as TRACE (for debugging) and a RENUMBER facility.

An innovating feature is the RUN+ Command which on encountering a GOTO < line number > will determine the absolute address of the line number and so speed up execution of a program.

The bad bits include a rather old fashioned line editor and a limit of 127 characters for strings.

Another shortcoming is the fact that although the Comx has a generous list of error codes, these are displayed as cryptic number displays which sends one reaching for the manual to decode them. With memory so cheap, its difficult to understand why the Comx cannot be more forthcoming in this area.

Cassette Interface

The cassette interface performed satisfactorily with the tones on tape being reproduced over the Comx's internal speaker. One shortcoming in this area was the fact that there was no on screen indication as to whether or not the loading of a tape was in progress – this make the initial setting of the recorder's volume control more difficult than usual.

The Manual

The manual supplied with the Comx was a well produced spiral bound tome that clearly explained the function of the machine as well as a number of program listings designed to demonstrate the capabilities of the machine.

As mentioned above, a major omission was the lack of any technical details of the machine's hardware.

Summing Up

In order for the Comx to gain a foothold in the market, potential customers will want to be sure of both hardware and software support. As we have already stated, the fact that the machine is based on the 1802 processor is a distinct handicap when it comes to software support.

Although from an examination of the manual it would appear as if there are plenty of plans for hardware expansion (there's a disk command resident) there is no sign of it here yet.

With the launch of the Microdrive about to give sales of the Spectrum a boost if any were required and, with the tremendous range of support for the Spectrum already in existence, it is difficult to see the Comx making a dent in Sinclair's sales.

If the Comx had reached the UK a year ago the story may have been different but, as with the Laser, it seems that now the Spectrum has too much of a strangle hold on this area of the market.

The only way to break the hold, would be to beat Sinclair at his own game and come into the market with a product that is not merely a look-alike but offers an undisputed advance in performance.

ZX SOUND BOARDS Part 2

Jim Cowie, D. O'Donnel and J. McAinsh continue their description of their PSG systems with details of software for the boards.

Last month's article described in detail the operation of the PSG itself and gave full circuit diagrams of both the Spectrum and ZX81 designs. Construction of the Spectrum board was also covered but lack of space prevented the PCB design for the '81 board being given. This month therefore, we'll begin with a description of this double sided board.

Putting it together

Because of the large number of interconnections involved in the ZX81 design, it was decided to go to a double sided PCB. This avoids the large number of links that would have been necessary with only a single sided board.

Construction is fairly straightforward, but be careful to ensure that all polarity sensitive capacitors and the ICs are connected the correct way round.

The edge connector can either be soldered directly to the board or can be connected via a length of ribbon cable.

The board is really in two sections and if you want to, you can leave out the amplifier

PARTS LIST

Resistors (¼ watt, carbon)

R1, R3	1k
R2	2k2

Capacitors
(all ceramic except 1 MicroFarad)

C1, C3	100n
C2	1n
C_d	10n
(decoupling)	1u

Integrated Circuits

IC1	AY-3-8910
IC2	74LS27
IC3	74LS00
IC4	74LS86
IC5	74LS368
IC6, 7	74LS74

Diode

D1	1N 4148

AUDIO AMPLIFIER SECTION

Resistors

R4	1k
VR1	10k preset potentiometer (carbon)

Capacitors

C4	300p
C5	2u2
C6	47n
C7	250u
C8	10u
C9	100n

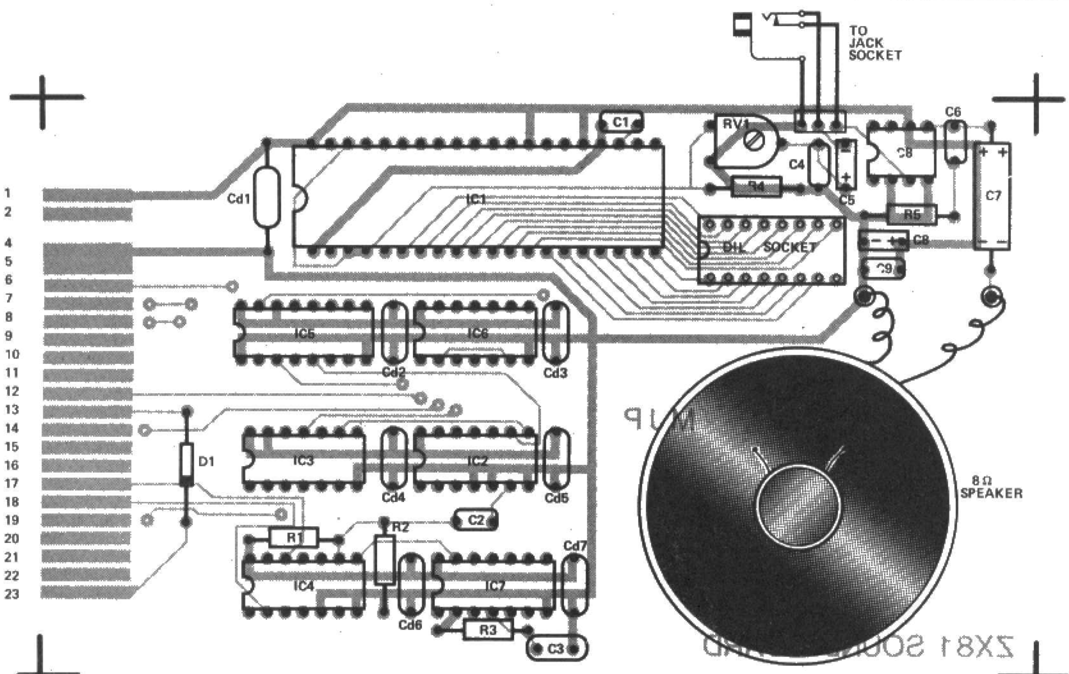
Integrated Circuit

IC8	LM386

Miscellaneous

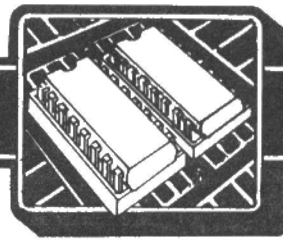
J1	switched jack socket
Speaker	8 ohm, 0.3 watt

0.1" veroboard, approximately 3.75" x 5", 2 x 23 way edge connector, short piece of ribbon cable.



NOTE: IF PCB NOT PLATED THROUGH, SOLDER COMPONENT LEADS THIS SIDE AS WELL AS SOLDER SIDE, WHERE NO COMPONENT IS USED LINK TOP TO BOTTOM WITH WIRE THUS:–

Figure 1. The ZX81 Sound Board's overlay. Note only the lower tracks of this double sided PCB are shown. Readers requiring a copy of the foil patterns, which have not been shown due to lack of space, can obtain them by sending a lage SAE to our offices.



part and simply feed the PSG audio output directly to an external amplifier, e.g. the monitor input on your cassette recorder.

A sixteen pin DIL socket in conjunction with a DIL header plug gives access to the 8-bit ports A and B.

Having completed the board, carefully check all components against the overlay and, if all is well, start experimenting with the software.

Programming the PSG

The PSG is an extremely powerful component so the best way to get to know it and how to run it from your ZX81 is to try things out. Because the generator is memory mapped, when you write (POKE) or read (PEEK) certain memory locations the value you write is sent to the internal logic of the PSG or read from the internal registers.

As already mentioned (8192) is the latch register address. We POKE this with a register number to tell the PSG to use this register for any data reads or writes. (8448) is used to read or write data. When we POKE a value to this address it is written in the selected register.

The following piece of code turns off all sound. It is advisable to have this somewhere in your program. Then if the program halts leaving the PSG still at full blast you can GOTO this statement and turn it off.

Turn off PSG

```
POKE 8192,8     (Select reg. 8)
POKE 8448,0     (turn it off)
POKE 8192,9     (select reg. 9)
POKE 8448,0     (turn it off)
POKE 8192,10    (select reg. 10)
POKE 8448,0     (turn it off)
```

For an easier way of entering the memory locations you can hold them as variables as in the next example. This program is based on an example given in the PSG guide. Timing can be a tricky problem from BASIC and some trial and error is required when writing programs.

Test Program

Finally, a test program which allows you to explore the possibilities of the PSG. A 16 element array is set up, in which each element corresponds to one of the 16 internal PSG registers. In the program the array is first set to zero then printed on the screen together with various options. You can alter the array and when finished enter –1 to get out of the alter routine. Enter P to print the new array in order to verify it. Enter R to load the array into the PSG registers. Enter S to stop the sound. When you wish to return to BASIC enter E to exit program.

Once a suitable sound has been set up you can add it as a subroutine to your own program. For example in a submarine game a continuous sonar beep could run on one channel (no processing required after it is set up) leaving two other channels for sirens and explosions.

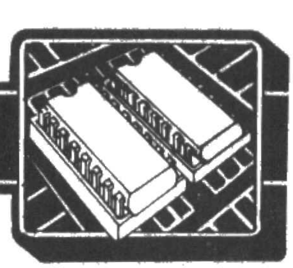
Now that you are aware of what the sound bard can do – a final word. If you have a model railway, why not use the input/output ports to control it? With suitable interfaces these could alter signals, open level crossings and sense the position of trains. This would allow whistles and other sounds to be produced at the correct moment.

```
****** RACE CAR SOUND EFFECT ******

  1  REM RACE CAR SOUND            !
  2  FAST                          !
  5  LET A= 8192                   ! set up POKE addresses
  6  LET B= 8448                   !
 10  POKE A,3                      ! set tone period on channel B
 12  POKE B,15                     !
 14  POKE A,7                      ! enable channels A & B
 16  POKE B,60                     !
 18  POKE A,8                      ! max amplitude on channel A
 20  POKE B,15                     !
 22  POKE A,9                      ! put channel B amplitude
 23  POKE B,10                     ! to a lower value
 24  LET F1= 2816                  !
 25  LET F2= 1024                  !
 26  LET F3= 4                     ! this value sets the frequency
 27  GOSUB 125                     ! jump in the sweep effect
 28  LET F1= 2304
 29  LET F2= 768
 30  LET F3= 4
 32  GOSUB 125
 33  LET F1= 1536
 34  LET F2= 512
 35  LET F3= 2
 36  GOSUB 125
 40  POKE A,8                      ! turn channel A off
 42  POKE B,0
 44  POKE A,9                      ! turn channel B off
 46  POKE B,0
 48  POKE A,10                     ! turn channel C off
 50  POKE B,0
 52  STOP
125  FOR I=F1 TO F2 STEP -F3       ! loop sequence
126  LET J=INT(I/256)              ! gives reg.1 value
128  LET K=I-256*J                 ! gives reg.0 value
130  POKE A,1                      ! select coarse tune A
132  POKE B,J
134  POKE A,0                      ! select fine tune A
136  POKE B,K
138  NEXT I
140  RETURN
```

```
****** TEST PROGRAM ******

                                                     Comments
 10  REM TEST PROGRAM FOR P.S.G.                     !
 20  LET L = 8192
 30  LET D = 8448
 40  DIM A(16)                                       ! initialise a 16 element
 50  FOR I= 1TO 16                                   ! array to zero
 60  LET A(I)= 0
 70  NEXT I
 80  GOSUB 1000                                      ! print initial array
 90  SCROLL
100  PRINT "R TO RUN       A TO ALTER"               ! print out options
110  SCROLL
120  PRINT "S TO STOP SOUND"
130  SCROLL
140  PRINT "P TO PRINT ARRAY TABLE"
150  SCROLL
160  PRINT "E TO EXIT PROGRAM"
170  SCROLL
180  INPUT R$                                        ! input option
190  IF R$ = "R" THEN GOTO 250                       ! jump to appropriate routine
200  IF R$ = "A" THEN GOTO 300
210  IF R$ = "S" THEN GOTO 400
220  IF R$ = "P" THEN GOSUB 1000
230  IF R$ = "E" THEN STOP
240  GOTO 90
250  FOR I= 1 TO 16                                  ! run routine
260  POKE L, I-1
270  POKE D, A(I)
280  NEXT I
290  GOTO 90
300  SCROLL                                          ! alter routine
310  PRINT "ALTER WHICH REGISTER ?"
320  SCROLL
330  PRINT "REMEMBER -1 ABORTS ALTER ROUTINE"
340  INPUT I
350  IF I < 0 THEN GOTO 90
360  SCROLL
370  PRINT "WHAT VALUE ?"
380  INPUT A(I+1)
390  GOTO 300
400  POKE L, 8                                       ! stop sound routine
410  POKE D, 0
420  POKE L, 9
430  POKE D, 0
440  POKE L, 10
450  POKE D, 0
460  GOTO 90
```



Spectrum Sound

The techniques for programming the Spectrum board are slightly different from those described for the '81 and make use of the Spectrum's OUT command.

Assuming all is well with the board, enter the following program to begin testing of the system.

```
10 OUT 189,0     (Pointer to register 0)
20 OUT 191,255 (Load Tune A)
30 OUT 189,7     (Point to register 7)
40 OUT 191,254 (Enables A to O/P)
50 OUT 189,8     (Point to register 8)
60 OUT 191,15   (Select O/P amplitude)
```

and RUN.

You will notice that unlike the SPECTRUM bleeper, the tone is continuous. For this reason it is useful to have the following programme available to reset the registers and turn off PSG.

```
9998  STOP
9999  FOR n = 0 TO 14: OUT 189,n:
        OUT 191,0: NEXT n
```

You can then RUN 9999 to turn the PSG off. Also, when experimenting you may produce a sound that you wish to record for future use only to find that it comes out differently because there was something in a register from previous experiments of which you were unaware.

An illustration of what sounds can be achieved can be heard with the following programmes:-

E&CM

```
10 REM  DYNAMIC SOUND EFFECTS
20 OUT 189,7 : OUT 191,248
30 OUT 189,8 : OUT 191,15
40 OUT 189,0
50 FOR A = 0 TO 128 : OUT 191,(A -64*INT(A/64))
60 OUT 191,128-A: OUT 191,A: NEXT A : OUT 191,0 : PAUSE 50
70 FOR B = 1 TO 10 : FOR A = 0 TO 255 STEP 10: OUT 191,
   A:NEXT A:NEXT B
```

```
10 REM RALLING BOMB
20 OUT 189,7: OUT 191,254: FOR A= 0 TO 255: OUT 189,
   8:OUT 191,INT(A/16)
30 OUT 189,0: OUT 191,A: NEXT A
40 DATA 6,13,7,7,8,16,9,16,10,16,12,20,13,0
50 FOR Z = 1 TO 7: READ A: READ B: OUT 189,A :
   OUT 191,B: NEXT Z
```

```
10 REM HELICOPTER
20 DATA 1,10,3,9,6,31,7,228,8,12,9,12,13,13
30 FOR Z = 1 TO 7: READ A: READ B : OUT 189,A:OUT 191,
   B: NEXT Z
```

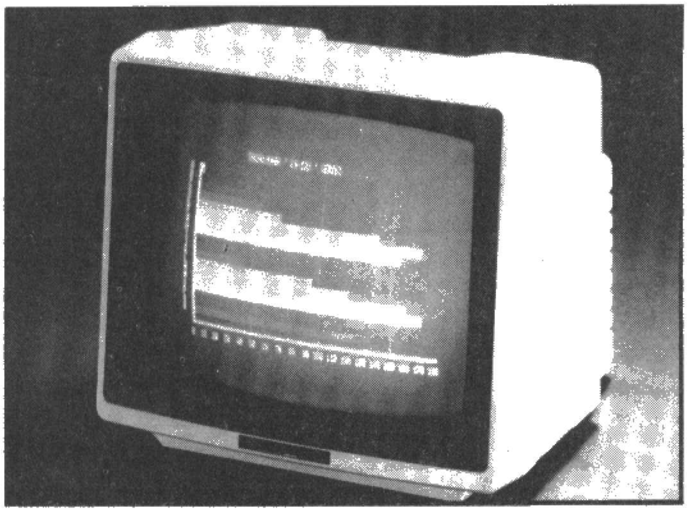
```
10 REM EXPLOSION
20 DATA 6,31,7,7,88,16,9,16,10,16,12,20,13,0
30 FOR Z = 1 TO 7: READ A: READ B: OUT 189,A: OUT 191,'
   B: NEXT Z
```

Finally the following programme can be used to develop your own sounds by entering the desired data into the PSG registers and displaying the contents of all the registers so that it can be seen at a glance the data that is producing the sound being heard.

```
10 INPUT"REGISTER"; R
20 INPUT "DATA"; D
30 CLS
40 OUT189,R : OUT 191,D
50 FOR N= 0 TO 13: OUT 189,N:PRINT AT (N + 1),3;N:PRINT
   AT (N+ 1),10; IN 189: NEXT N
50 IF R = > 15 THEN GOTO 9999: GOTO 10
9998 REM RESET ALL REGISTERS TO ZERO
9999 FOR N = 0 TO 14: OUT 189,N: OUT 191,0: NEXT N
```

MONITOR SURVEY

Nick Clare has looked at a number of popular monitors. Read his report to find out what's available and from whom.



The importance of using a good quality monitor capable of producing text lines greater than 30 characters or of generating all but the 'chunkiest' of graphics has been emphasised elsewhere in this issue (see the introduction to the ZX Video Techniques article). For many people, upgrading their computer to one of the more powerful of today's machines will only serve to highlight the limitations of a standard TV set fet from a UHF signal when it comes to hi-resolution work.

The only solution to these problems is to use a monitor video signal or, for the best results with a colour display, to drive the monitor with an RGB signal.

This survey takes a look at a range of monochrome and colour monitors that range in price from about £80 (for a basic monochrome model) to units costing almost £500 (hi-res colour monitors).

Starting Point

When contemplating the production of a monitor, a manufacturer has a fundamental choice to make. Either they can produce a design from scratch or, alternatively, modify an existing TV chassis to produce a unit that is capable of both off air TV reception and of accepting either composite or RGB video signals.

The former route is likely to produce a design that is capable of higher ultimate quality although many purchasers will be tempted by the converted TV sets as effectively they are getting a colour TV set in to the bargain.

Specifications

When looking through the specification of a monitor the first and most obvious point for consideration is the compatability of the monitor and the computer with which it is to be used. With monochrome and colour monitors with a composite video input there is no difficulty – the composite video signal is a well standard. With colour monitors with RGB drive to situation is not nearly as clear cut.

The RGB video signal can be positive or negative going as can the sync signal – four permutations on the basic theme. In addition, some computers will output the sync signal, not as a seperate entity, but as part of the green video signal.

To complicate matters still further, some computer manufacturers produce video outputs that are not straightforward RGB drives but are encoded in a way peculiar to the particular computer. In these cases either a dedicated monitor is required or an interface to convert the computer's output to a standard drive waveform.

About the only way to ensure compatability is to try the monitor out on the system it is to be used with.

The only other area of a monitor's specification that is of major import is the ultimate resolution of which it is capable.

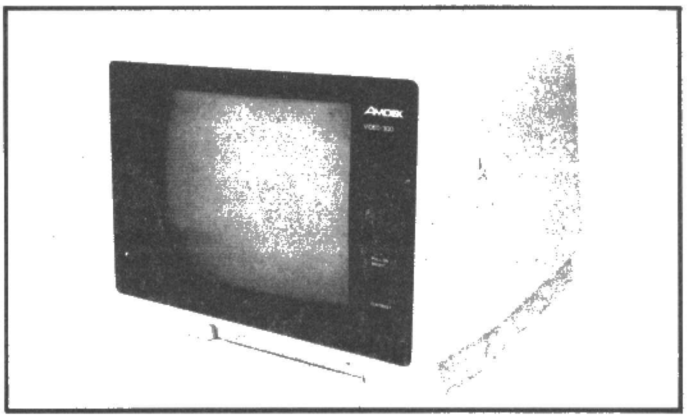
'Resolution' can be expressed both in terms of the number of lines a monitor is capable of resolving in both the horizontal and vertical directions (a measure of the tube's capabilities) or in terms of the bandwidth of the monitor (a measure of the performance of the monitor's electronic circuitry).

In the case of the number of lines resolved in the vertical direction, the limit will usually be imposed by the scanning format adopted by the computer and will be in the region of 300-400 lines. In the horizontal direction, the greater the number of lines quoted, the better the definition of display will be.

Other aspects of a monitor's performance quoted in a particular model's specification are of less importance but will include such items as figures for the display's linearity (the smaller the figure quoted the better).

Monochrome Monitors

We'll begin with a look at the various monitors available with **LB Electronics** model HM-910. This is a 9" green phosphor monitor with a standard 75R composite video input capable of accepting inputs in the range 0.5V – 2.0V pp. Bandwidth is quoted at 18MHz and resolution is 350 lines vertical and 700 lines horizontal. At £89.70 including VAT the unit is a good, high performance monitor.



The **Amdek 300** monitor distributed by Roland UK has a slightly larger 12" screen but offers a similar performance with an 18MHz bandwidth. User controls provided on the unit are brightness, contrast, horizontal and width, vertical height and linearity is also provided.

Price of the unit is yet to be fixed but is to be under £135 – contact Roland for the exact price.

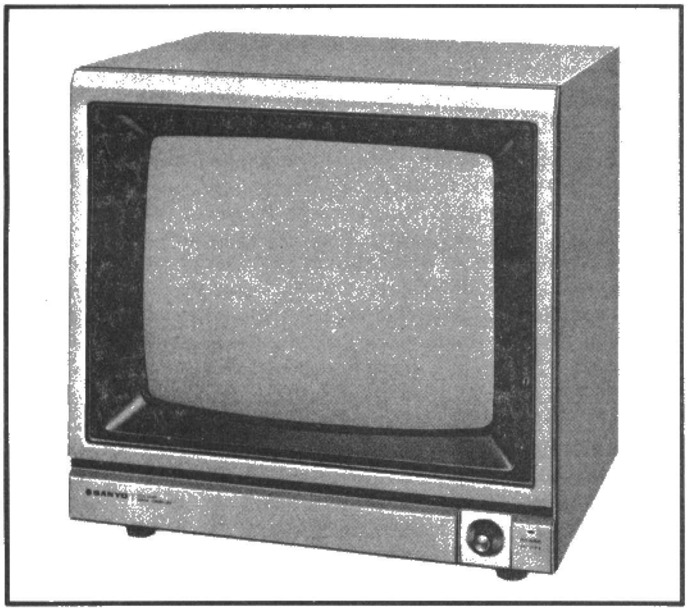
Kaga are another source of good quality monochrome monitors and can supply their hi-res K12 unit in a choice of either green or amber phosphor's at prices of £109 and £119 respectively.

Colour Convertibles

Newark Video Centre offers a range of modified Grundig TV sets that are capable of receiving either off-air TV signals or an RGB and sync input. The video signal is via a 6 pin DIN socket and features linear analogue inputs they mean, in theory, an infinite number of colours can be produced by the design.

The modifications are fully approved by Grundig and prices start at £275.54 for a 14½" set and go up to £720.87 for a 26" remote controlled set with stereo sound and Teletext.

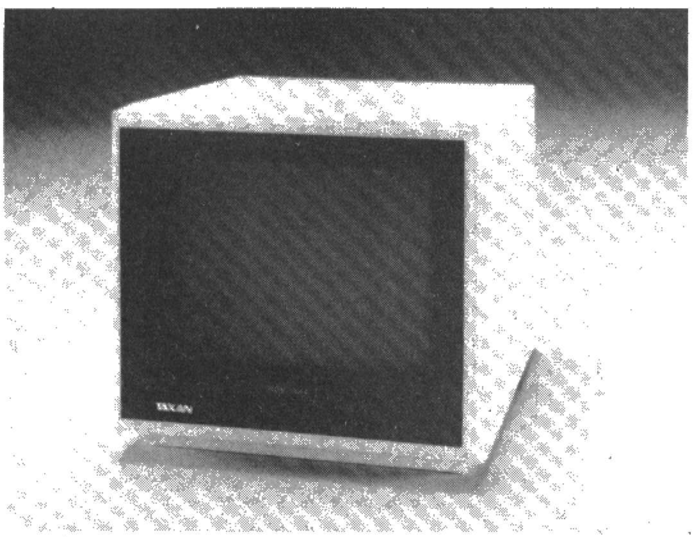
Portatel conversions are another company that undertake conversions. This company have chosen the Luxor range of sets to base their conversions.



These sets are fitted with Toshiba PIL tubes and, its claimed, that RGB resolution is limited only by the tube itself. Resolution is quoted by the very practical expedient of stating that the 14" monitor is capable of resolving lower case characters of the BBC micro in mode 0. Prices range from £249 for a 14" set to £499 for a 26" monitor.

Portatel also produces a rather unusual converted set designed for IBM computers. The RGB input is as those on the standard sets but the set features an NTSC colour decoder for use with the RF outputs of American computers.

Better Electronics also have a converted TV chassis that is compatible with the BBC micro. RGB input is again via a DIN plug and the set comes complete with a 1 year guarantee and even a wired main plug. The price is around £210 + VAT but contact Better for details.



Finally in this section, Electronequip can supply an RGB and Composite monitor at £226 + VAT. The monitor is a medium resolution design and features a bandwidth of 7MHz.

Dedicated Design

Cabel Electronic supply a range of purpose built RGB monitors that range from CE370A capable of resolving 430 pixels in the horizontal direction at a cost of £199.50 + VAT to the HR370 at £420 + VAT that is capable of resolving 640 pixels horizontal and has a bandwidth of 22MHz.

All the monitors are attractively designed and have all user controls and inputs on the front panel – this saves groping around the back of the monitor to change the brightness or unplug the unit.

Kaga, as well as their monochrome design produce a range of RGB monitor of which the K12R1 is a typical example.

One of the most popular RGB monitors is the Sanyo 3125NB. This is available in a number of different versions from a standard resolution model for just over £200 to the highest resolution model that costs approaching £450. This set can cope with all the visual RGB signal permutations by way of jumper wires within the monitor's case.

It also has a separate input for controlling the display of the cursor.

Finally mention should be made of the **Sword** range of monitors that range from a standard resolution 450 line model to a 400 line super resolution design. The Sword range of monitors feature a screen that is jet black rather than the dull grey of many designs. This adds impact to the display, emphasising the contrast between the background and any display.

LB Electronics
11 Hercies Road
Middlesex
CB10 9LS

Roland UK
Unit 6
Great West Trading Estate
983 Great West Road
Brentford
Middlesex
TW8 9DN

Kaga
Data Efficiency Ltd.
Finway Road
Hemel Hempstead
Herts
HP2 7PS

Newark Video Centre
108 London Road
Balderton
Newark
Notts
NG24 3AQ

Portatel Conversions
25 Sunbury Cross Centre
Sunbury-on-Thames
Middlesex

Better Electronics
58 Mill Road Avenue
Angmering Village
Sussex
BN16 4HS

Cabel Electronic
19 High Street
Tewkesbury
Gloucestershire
GL20 5AW

Sword Monitors
Corton Electronics Ltd.
Rockland Works
Eagle Street
Coventry
CV1 4GJ

Electronequip
36-38 West Street
Fareham
Hants
PO16 0JW

Understanding Your Computer

In part seven of the series, B. Boyde-Shaw looks at mass storage systems, from the lowly cassette to highly sophisticated devices such as EAROM.

Once a program has been written, fully tested and debugged, it's usual to want to save it for posterity, and for future use. The cheapest method of storage, at the moment, is the cassette tape, using a commercial audio cassette recorder. Unfortunately cassette recorders work beautifully when dealing with audio signals, and only the most acoustically aware of this world notice the small inaccuracies of tone, pitch or 'quality of sound' produced by your 'specially reduced offer' music centre. Unfortunately the computer belongs to the acoustically aware group and needs a reasonably clean signal, that has been properly transcribed, and is reasonably noiseless.

The fact that a recorder may have lost a few harmonics in passing, doesn't really affect a normal recording of Beethoven's 5th, Cliff Richard's 1005th, or even Spandau Ballet's latest, but if the computer's cassette read system sees a bad note, then the message is incomplete and the program, when RUN, will crash or grind to a halt, with the inevitable error message.

So what to do?

Following the general guide lines below should result in the minimum of trouble, provided a reasonably well made, unsophisticated cassette recorder is used – no stereo, no tone control, no AFC – just jacks and volume.

Loading Hints

When loading from a cassette recorder, adjust the volume to maximum (any other controls to about x) and type CLOAD "" (load "" on BBC). Press ENTER and play on the recorder then watch for an on screen indication that loading is taking place.

If after one minute nothing has happened stop and rewind the tape. Check carefully that a) the leads to and from the recorder are correctly wired and b) that there is a program on the tape – sorry for pointing out the obvious but it's surprising how often it's possible to try and load a blank tape.

If all is well, adjust the volume output of the recorder in small steps (try increasing the setting first) until a successful load can be obtained. When loading is successful mark the cassette's label with the volume setting shown on the recorder.

Saving Hints

The best tapes to use for saving programs are short (no longer than C15) tapes designed for the task. After connecting up the recorder wind on at leas two feet of tape to avoid the plastic leader at the start of the cassette.

Type in CSAVE "< prog name >" (BBC – save"") and put the cassette recorder into the record mode.

Press ENTER whereupon the screen should show that loading is taking place. When saving is complete, stop the recorder and reqind the tape.

It is a good idea to verify all tapes and if your computer has a verify facility, the verification procedure described in its manual should be followed before removing the program from the machine's RAM.

Disc storage

The disk storage system is a much more sophisticated method than the cassette system, and works a little like a record player, but here the records don't have continuous tracks.

Programs are stored on a disc, (record), using a disc drive unit, (record deck), which uses a read/write head, (stylus).

The system is relatively bug free and foolproof, but requires care in the storage of discs – not on the back of the TV set, and never bent or overheated. Treat them just like an expensive record.

The system is much quicker than cassette, (x100 for example), and like cassettes can be used inside a program for retrieving and storing data, but very much more efficiently. The disc system can be considered as an extension to a program, because of its undoubted ability to locate the required data nigh on immediately. Whereas the cassette recorder has to run past oceans of unwanted material before coming across the bit that's required.

A good disc system, not surprisingly, costs a lot more than a cassette recorder – say a factor of ten at least, but it's worth paying for if a lot of data is handled by a program.

Types of disc

There's an hierarchy of mass storage that seems to get longer and more complicated with each month that passes.

For example we have: cassette tape – floppy discs – hard discs – RAM discs – bubble memories – video discs – and last but not least – laser cards.

Floppy discs

Let's look at the floppy disc first, the most common form of disc storage for the home computer user's system. It can have a variety of names – floppies – diskettes – minidiscs, and be of two types – hard or soft. Soft floppies come in three sizes – 8 inch – 5.25 inch and 3.5 inch diameter. The 8 inch have about 70 odd tracks (uncontinuous individual ones), but are not used much now, so we will only consider the most popular size, the 5.25 inch. This size has 40 tracks, with each track divided into a number of sectors or blocks of 256 bytes each. The disc surface itself is covered with a magnetic material like a cassette tape, so that the 1s and 0s can be stored as required, but the density of storage available varies with the description of the disc.

Single sided single density, as the name implies, has information stored on one side only. These can store about 50K bytes. The single sided double density has been designed to store considerably more information, more blocks or sectors per track, up to 250K bytes per side.

Double sided discs obviously do it twice, one on each side, but are considerably more expensive, as is the hardware needed to operate them.

The disc is round and flexible and must always be kept in its cardboard cover, which has a central circular hole cut in it and somewhere on its diameter a slot, which allows the disc read/write head to access/store the required information.

The disc operating system, or DOS, positions the disc read/write head over the correct area of the disc, after rotating it to expose the required sector. By the way the head is positioned using a stepper motor and the drive unit a more normal motor, spinning the disc at 350 revs per minute, giving a data transfer rate of about 250K per second – compare that to the normal cassette tape system of around 1.2K per second.

Hard discs

Hard discs now have a standard, the Winchester 5.25 inch, (the name Winchester is only a code name used by IBM for one of its products in this field). The hard disc drive motor rotates 10 times faster than the floppy type, with 625 byte per second data transfer rate, giving an access time as fast as 60ms. The machinery works in the same way as a floppy system, but is an extremely high precision piece of engineering, with the read/write head running much closer to the disc, (that's why it has to be hard you see). It's possible to get as much as 5 and 10 Megabyte storage capacity and all for £1500. It looks as if we

are stuck for the time being with floppies and paying anything up to £600 for our disc system.

ROM and RAM

In this area of information storage we are considering more the devices a computer uses and not the programmer.

There are two main types of memory, volatile and non-volatile. Volatile meaning that the information you have stored is lost when the device's power supply is switched off. Non-volatile means that even if the power supply is disconnected the information is not lost. So obviously cassette tape and discs are examples of non-volatile memory devices.

A further example is the ROM device, standing for Read Only Memory. Once programmed with information the ROM memory is set for life, unless something drastic happens, such as finding itself surrounded by a strong magnetic field, which might make it lose all its memory for ever, a kind of chronic benign computerised amnesia. So keep those expensive modules and cartridges away from the back of your television set! The ROM is used by the computer to store the instructions for its operating system, and it's where you'll find its language stored too, i.e. BASIC or FORTH for example.

ROMs can also be used to add to a computer's operating system, as in the previously mentioned modules and cartridges, and for transferring data, etc., into RAM for program runs. Some sophisticated examples of ROM are the BBC wordprocessor packages – VIEW and WORDWISE, speech synthesizers for the BBC, TEXAS and VIC20 now appearing on the market.

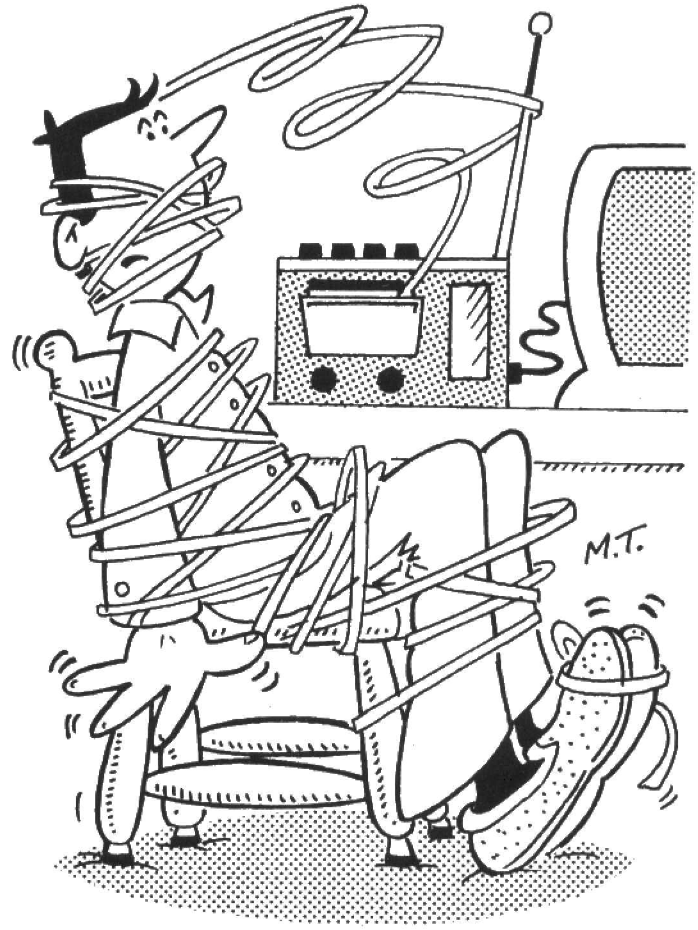
To confuse matters further there are a number of other types of ROM. A PROM is a Programmable Read Only Memory, which means that instead of buying from a manufacturer a preprogrammed ROM, you buy one in a blank state and program it yourself to suit your own requirements. For example to illustrate the main difference between ROM and PROM consider a set sequence of digital signals such as the morse code which could be programmed onto ROM and sold as a package. But consider the program for checking the speed and petrol consumption in one of the more modern cars, this could only be programmed by the designer of the motor car.

ROM and PROM once programmed are fixed and cannot be altered, but there is a chip called an EPROM which can be programmed, then later as the program develops, be erased and reprogrammed. One future use of EPROM could be the interchange of messages, ideas, programs etc., between computer users, in the same way that tapes and discs are used now. The advantages would lie in lower transport costs, little likelihood of transit damage, and less vulnerability to attack by stray magnetic fields and high temperatures.

To erase an EPROM, an ultra-violet light source is required, and then it can be reprogrammed in the usual fashion, but there is yet another type of ROM called an EAROM, which stands for Electrically Alterable Read Only Memory, and is both erasable and programmable by electrical means, which would seem to bring us to the RAM. The difference between ROM and RAM is mainly in circuit design and the way that the signal is written into the memory cell.

RAM stands for Random Access Memory and is the commonest type of user memory in micro computers, but being of the volatile type, loses it memory on switch off.

The unit of data is the Kilobit, (as also is ROM), equalling 1024 bits of information, that is 10 to the power of 2, and a byte equals 8 bits. As computer scientists do their mathematics in binary the modules of RAM are usually 2K, 4K, 8K, 16K, 32K etc., the most common multiples being 4K, 16K and 32K for home computers. But 16K of RAM usually means 16 Kilobytes of RAM, that is 16 x 1024 x 8 or 131072 bits of information. When talking about integrated circuits, a 16K RAM IC would be referring to the number of storage places, bits, so would only have 16384 bits, therefore to build a 16K RAM computer, you would need 8 x 16K RAM ICs, or 32 x 4K.



Problems — what problems?

To the future

Laser cards are what dreams are made of, but then so was flight. The laser card consists of a card about the size of a credit card covered in silver halide particles, which is read by a laser beam. It will hold about 5 Megabytes, which would allow a goodly number of paperbacks to be stored, or about a million words in total – Dickens on credit card – so to speak, or perhaps carded Cartland.

Video discs are what dreams come true are made of. They are on the market now, but not used for home computer storage, and are used for storing and replaying films instead of using video tapes. At the moment they are very difficult to write to, as lasers are required, but they are easy to read from. They could be used to store anything up to 4000 Megabytes.

Bubble memories are a non-volatile form of RAM, expected to hold about 4 Megabytes of memory, again a little too far off for the home computer enthusiast.

Maybe the day is not too far distant when tapes and discs will be a thing of the past, only seen in college museums, and all user memory will be contained in something the size of a cigarette packet, in the Megabyte range, and as cheap as a cassette tape is today.

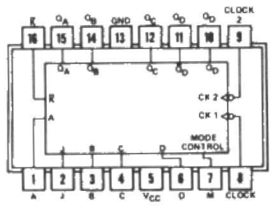
But isn't that where user memory all started – making a note on the back of a cigarette packet?

E&CM

INTEGRATED DATA

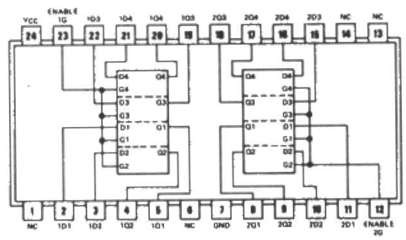
Continuing our 'cut out and keep' TTL Data series.

99



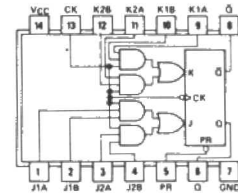
4-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTERS

100



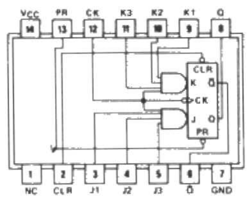
8-BIT BISTABLE LATCHES

101



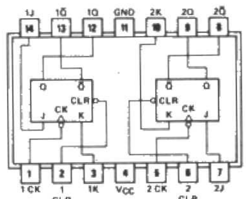
AND-OR-GATED J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET

102



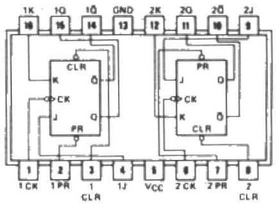
AND-GATED J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

103



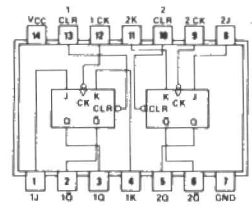
DUAL J-K NEGATIVE-EDGE-TRIGGERED FLOPS WITH CLEAR

106



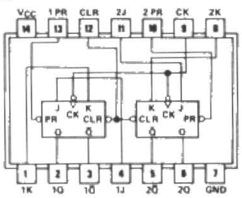
DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

107



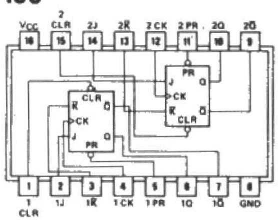
DUAL J-K FLIP-FLOPS WITH CLEAR

108



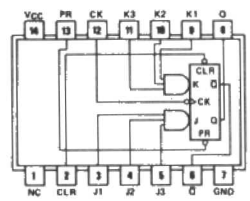
DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET, COMMON CLEAR, AND COMMON CLOCK

109



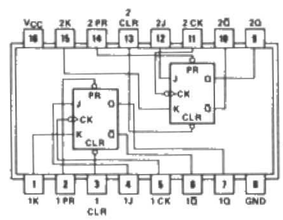
DUAL J-K POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

110



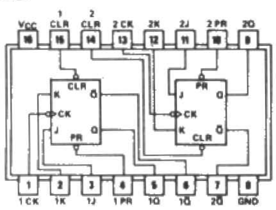
AND-GATED J-K MASTER-SLAVE FLIP-FLOPS WITH DATA LOCKOUT

111



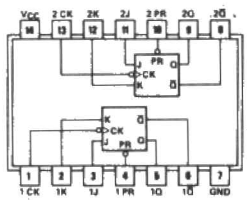
DUAL J-K MASTER-SLAVE FLIP-FLOPS WITH DATA LOCKOUT

112



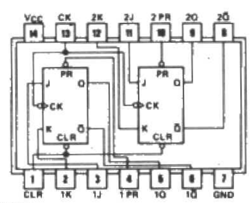
DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

113



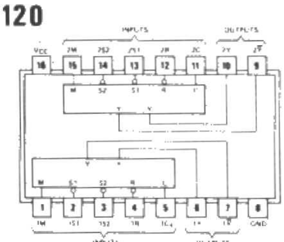
DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET

114



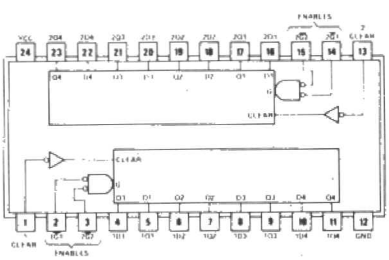
DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET, COMMON CLEAR, AND COMMON CLOCK

120



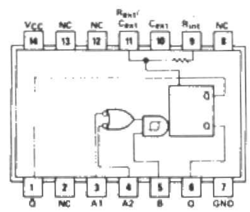
DUAL PULSE SYNCHRONIZERS/DRIVERS

116



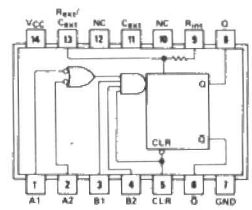
DUAL 4-BIT LATCHES

121



MONOSTABLE MULTIVIBRATORS

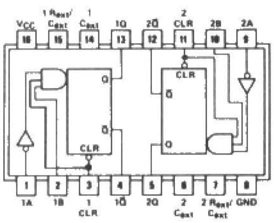
122



RETRIGGERABLE MONOSTABLE MULTIVIBRATORS WITH CLEAR

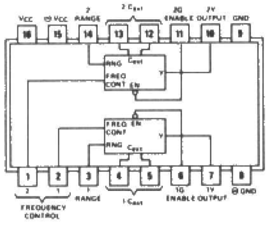
INTEGRATED DATA

123



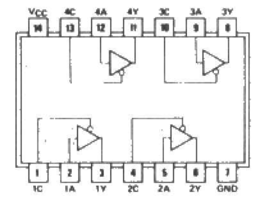
DUAL RETRIGGERABLE MONOSTABLE MULTIVIBRATORS WITH CLEAR

124



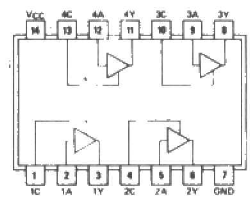
DUAL VOLTAGE-CONTROLLED OSCILLATORS

125



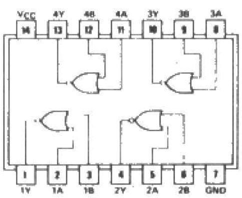
QUADRUPLE BUS BUFFER GATES WITH THREE-STATE OUTPUTS

126



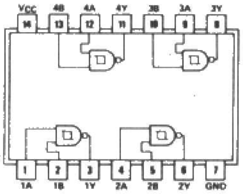
QUADRUPLE BUS BUFFER GATES WITH THREE-STATE OUTPUTS

128



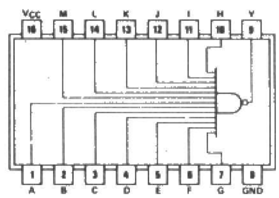
50-OHM LINE DRIVER

132



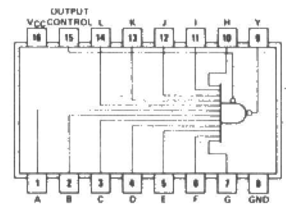
QUADRUPLE 2-INPUT POSITIVE-NAND SCHMITT TRIGGERS

133



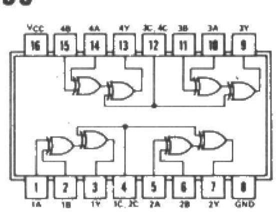
13-INPUT POSITIVE-NAND GATES

134



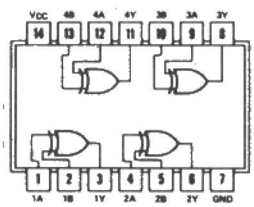
12-INPUT POSITIVE-NAND GATES WITH THREE-STATE OUTPUTS

135



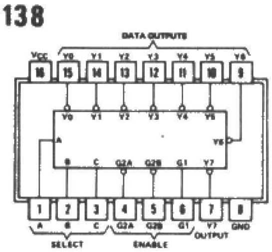
QUAD EXCLUSIVE-OR GATES

136



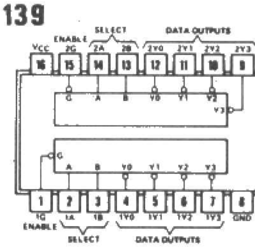
QUAD EXCLUSIVE-OR GATES WITH OPEN COLLECTOR OUTPUTS

138



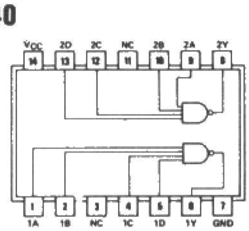
3-TO-8 LINE DECODERS/MULTIPLEXERS

139



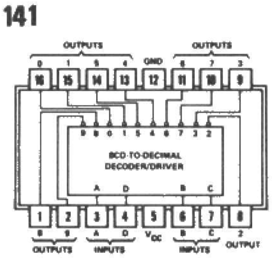
DUAL 2-TO-4 LINE DECODERS/MULTIPLEXERS

140



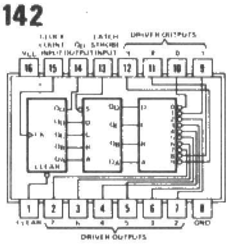
DUAL 4-INPUT POSITIVE-NAND 50-OHM LINE DRIVERS

141



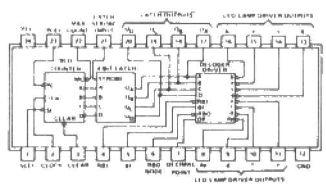
BCD-TO-DECIMAL DECODER/DRIVER

142



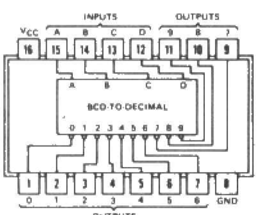
COUNTER/LATCH/DECODER/DRIVER

143



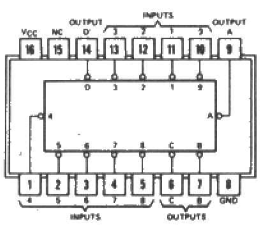
COUNTERS/LATCHES/DECODERS/DRIVERS

145



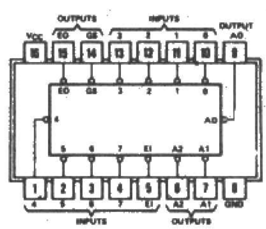
BCD-TO-DECIMAL DECODERS/DRIVERS FOR LAMPS, RELAYS, MOS

147



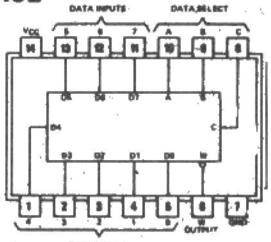
10-LINE DECIMAL TO 4-LINE BCD PRIORITY ENCODERS

148



8-LINE-TO-3-LINE OCTAL PRIORITY ENCODERS

152



1-OF-8 DATA SELECTORS/MULTIPLEXERS